Improving Neural Question Answering with Retrieval and Generation

Patrick Simon Hampden Lewis

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
of
University College London.

Department of Computer Science
University College London

July 7, 2022

I, Patrick Simon Hampden Lewis, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

Text-based Question Answering (QA) is a subject of interest both for its practical applications, and as a test-bed to measure the key Artificial Intelligence competencies of Natural Language Processing (NLP) and the representation and application of knowledge. QA has progressed a great deal in recent years by adopting neural networks, the construction of large training datasets, and unsupervised pretraining. Despite these successes, QA models require large amounts of hand-annotated data, struggle to apply supplied knowledge effectively, and can be computationally expensive to operate. In this thesis, we employ natural language generation and information retrieval techniques in order to explore and address these three issues.

We first approach the task of Reading Comprehension (RC), with the aim of lifting the requirement for in-domain hand-annotated training data. We describe a method for inducing RC capabilities without requiring hand-annotated RC instances, and demonstrate performance on par with early supervised approaches. We then explore multi-lingual RC, and develop a dataset to evaluate methods which enable training RC models in one language, and testing them in another.

Second, we explore open-domain QA (ODQA), and consider how to build models which best leverage the knowledge contained in a Wikipedia text corpus. We demonstrate that retrieval-augmentation greatly improves the factual predictions of large pretrained language models in unsupervised settings. We then introduce a class of retrieval-augmented generator model, and demonstrate its strength and flexibility across a range of knowledge-intensive NLP tasks, including ODQA.

Lastly, we study the relationship between memorisation and generalisation in ODQA, developing a behavioural framework based on memorisation to contextualise the performance of ODQA models. Based on these insights, we introduce a class of ODQA model based on the concept of representing knowledge as question-answer pairs, and demonstrate how, by using question generation, such models can achieve high accuracy, fast inference, and well-calibrated predictions.

Impact Statement

This thesis investigates how to improve the abilities of NLP models to leverage knowledge expressed in textual corpora. As the amount of knowledge humans generate about the world grows, it becomes increasingly important to build systems that allow us to efficiently leverage this knowledge. We directly study Question Answering models which are designed to serve the information needs of humans.

In addition, as NLP-equipped systems become ubiquitous features in our lives, it is vital that we construct models which have a strong command over factual knowledge, whilst also providing mechanisms for updating and controlling their knowledge, and presenting evidence to support their predictions. The work presented in this thesis enables more accurate QA models, but also improves our understanding of how different QA models operate, and what their limitations are. We also develop methods for widening the reach and applicability of QA technology, into settings and languages without annotated data, and present efficient and accurate QA models which require less-demanding hardware.

Finally, whilst our focus in this thesis is on QA models, we note that the progress we make on QA may also benefit other knowledge-intensive NLP domains, often with important potential societal benefits beyond QA, such as automated fact-checking. This work presented in this thesis has led to a number of publicly available research datasets and resources for the QA community, as well as open-sourced code and models, and has generated publications at the leading NLP and Machine Learning venues: ACL, EMNLP and NeurIPS.

Acknowledgements

First and foremost, I would like to thank Sebastian Riedel for his guidance, patience, kindness, good humour and inexhaustible enthusiasm and energy. Sebastian is really one-of-a-kind, and I am so grateful to have had the privilege of him being my primary advisor. I must also thank my secondary advisor, Pontus Stenetorp. Pontus has taught me so much about the world of research, and his hands-on and deeply caring, conscientious approach to supervision is an inspiration.

I also want to thank my examiners, Prof. Emine Yilmaz and Prof. Ivan Titov. Both examiners deeply engaged in my thesis, asked insightful questions, and I enjoyed discussing my work with them greatly. I am grateful for their post-viva feedback, which has strengthened the thesis.

I want to acknowledge all my co-authors, collaborators, colleagues and friends at UCL, FAIR, and beyond. There are too many of you to mention, but in particular, a special thank you to Fabio Petroni, Tim Rocktäschel, Max Bartolo, Ethan Perez, Barlas Oğuz, Pasquale Minervini, Yuxiang Wu, Aleksandra Piktus, Heinrich Küttler, Linqing Liu, Johannes Welbl, Douwe Kiela, Wen-tau Yih and Holger Schwenk. Without your help, friendship and guidance, this thesis would not have been possible.

Thank you to my mentors prior to my PhD journey. Thank you to Charles Conway, Angela Drew, Sarah Williams, Tracy Muller, Andy Wilson and Helen Hynd, for your encouragement and teaching me how to think. Thank you to Prof. Andy Rice for igniting my interest in Computer Science, and your support, belief and enthusiasm. Thank you to Prof. Rod Jones for teaching me to develop intuitions to understand complex phenomena and the value of being approximately correct. Finally, Thanks to Prof. Johnathan Goodman for encouraging and challenging me, and giving me a taste of how rewarding technical academic research can be.

Lastly, I must thank my family and friends. Thank you to my brother Rich, for being an inspiration. Thank you to my parents for their never-ending support, and raising me with a love of learning and thinking. And thank you to Joanna, for sharing the dizzy highs and lows of my PhD journey, and being the best team-mate I could ask for, especially during long years of lockdown and "Work-from-Bedroom".

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Natural Language Processing (NLP) and the representation and application of knowledge are two of the central goals of Artificial Intelligence (AI) research (Poole et al., 1997; Russell and Norvig, 2003; Luger, 2008). The field of Question Answering (QA) spans these two long-standing AI challenges. In order to answer questions posed in natural language, one must first be able to process and understand natural language itself, *and* be able to capture, represent and recall sufficient knowledge to formulate a satisfactory answer. Indeed, not long after computer science emerged as a concrete discipline, it was recognised that asking a computer questions was a powerful and general assessment of its intelligence (Turing, 1950). Developing models which can answer questions well, on a broad set of topics, requires mastery of natural language, a comprehensive knowledge of the world, and is a promising and worthwhile pursuit in AI research (Ferrucci et al., 2010).

Question Answering is also of great practical value. Humans have created and stored on the order of zettabytes of data, orders of magnitude more than any one human could ever observe (Reinsel et al., 2018). However, this knowledge is only valuable if we build systems that enable us to efficiently access and use it. Questions and answers are arguably the most common natural way that humans request and transfer knowledge from one another. In addition, many people lack the computer literacy skills to operate complex, non-natural-language-based interfaces, especially in less economically developed countries (Schwab, 2019). Thus, providing QA capabilities to computers allows for the most natural, and thus perhaps most appropriate, inclusive and equitable access to the knowledge they store. Beyond this societal importance, QA systems have also been recognised for their commercial applications, and natural language interfaces, including QA capabilities have been widely adopted in systems such as the Google Assistant, Siri, Alexa and others.

Question answering involves formulating answers to questions using knowledge in a *knowledge source*. There are many modalities in which knowledge can exist, such as in images or structured knowlegebases (KBs). The majority of the world's knowledge currently exists in unstructured formats (Gandomi and Haider, 2015), and even for structured KBs, unstructured textual data is still present in the form of descriptions and metadata. Textual data is also comparatively easy for humans to produce relative to structured formats, and it is common for structured data to be created from textual data. Thus, in this thesis, we focus on QA over *textual* knowledge sources. In the general case, we shall consider asking questions given the entire text of Wikipedia as a knowledge source. This task is referred as *open-domain question answering* (ODQA) in the literature.

The challenge and breadth of the ODQA task can be illustrated by considering the two ODQA questions in Figure 1.1. Example A is a typical *information-seeking* question, as entered into a search engine, whereas Example B is a trivia question, designed to *test* for knowledge and intelligence.

The first step in ODQA is to understand the meaning and intent of the question. Despite its apparent simplicity to the human eye, Example A's question requires some sophisticated language processing and inference skills. "Reba" must be identified as a named entity, likely a person or group of people, despite its relative word rarity. It also requires parsing the noun phrase "does he love me", overcoming the syntactically unusual structure and lack of punctuation cues. Inferences are required that "does he love me" is some kind of musical work involving singing, and "reba" is responsible for singing in it. Finally, we must infer the answer is going to be an

Question: who sings does he love me with reba
Answer: Linda Davis

Evidence: "Does He Love You" is a song written by Sandy Knox and Billy Stritch , and recorded as a duet by American country music artists Reba McEntire and Linda Davis. It was released in August 1993 as the first single from Reba's album Greatest Hits Volume Two. It is one of country music's several songs about a love triangle. [SOURCE]

(a)

Question: What is the name given to copper bars arranged in a cylinder, insulated from each other which rotates to connect each section of the armature in turn (in a motor) or to the external current (in a dynamo)?
Answer: Commutator

Evidence: [...] A commutator is a rotary electrical switch in some motors that supplies current to the rotor. It consists of a cylinder composed of multiple metal contact segments on the rotating armature [...] A commutator periodically reverses the current direction in the rotor windings [...] [SOURCE]

(b)

Figure 1.1: Examples of Open-domain questions, their answers, and sufficient textual evidence to answer them which can be found in Wikipedia for a) NaturalQuestions (NQ, Kwiatkowski et al., 2019) and b) TriviaQA (TQA, Joshi et al., 2018)

entity capable of singing, such as a person or band. Example B's question's difficulty is more apparent – it is long, contains complex co-ordination structure, spacial reasoning and complex terminology, and requires the inference that the answer is going to be the name of a piece of machinery.

When questions have been processed, we must then consult the knowledge source to find potential answers. In general this presents a *large-scale* search problem over typically millions of paragraphs that may provide an answer. In order to surface the evidence passage shown for Example A, a model must realise that the questioner has made a mistake, and the song is actually called "Does he love *you*", not "Does he love *me*". For Example B, the evidence passage is full of complex terminology, and fairly low surface-level word overlap with the question.

Finally, once evidence has been assembled, a model must formulate an answer. In these examples, this can be achieved by extracting a span of text. This task – extracting an answer to a question from a paragraph of text – is referred to as *reading comprehension* (RC). RC is by no means trivial. For Example A, there is no explicit statement that Linda sang the song. There are distracting plausible entities of the appropriate semantic type (Sandy and Billy), which must be discounted. Moreover, to arrive at the correct answer, implicative reasoning and a strong grasp on the meaning of "duet" is required – we must infer that "recorded as a duet" means that the two artists must have both sung. For Example B, sophisticated language understanding is needed to infer the equivalence between the descriptions in the question and evidence, e.g. equating "copper bars" with "metal contact segments".

1.1 Aims and Themes

In recent years, neural network models have swept the research landscape, and it has become common to parameterise every component of text-based QA systems with a neural network, typically with supervised learning using a large dataset. The representational power of neural language systems has recently been greatly improved by the development of unsupervised pretraining of larger models on large corpora, using objectives derived from denoising and language modelling. These models have established a new NLP paradigm, and represent a distinct step-change in i) our natural language understanding and representational abilities, but also ii) our natural language generation capability. Techniques that were possible in theory, but impractical due to a lack of model competency, have now abruptly become feasible. For example, *parametric knowledge* – relational knowledge that pretrained models encode in their parameters from their corpora – enables new kinds of factual retrieval paradigms. As a consequence, this thesis is largely a response to the emergence of these new pretrained models and techniques: what they enable for text-based QA, to what extent, and how to make the most out of them.

Aims: Our aims are to extend the general abilities of neural QA models, and explore and assess solutions intended to address their known flaws. For example, neural RC models are data hungry, which limits the domains and languages that they can be applied in. ODQA models suffer similar issues as RC models, with the additional challenge of large-scale retrieval from their knowledge sources, as well as suboptimal and inflexible pipe-lined architectures and slow, expensive inference.

Themes: We identify *retrieval* and *generation* as particularly powerful and versatile tools in modern NLP, and apply them to ameliorate some of these issues at the frontier of text-based QA. The following highlights key themes and techniques that will occur throughout this thesis.

Retrieval As our natural language representation abilities improve, so do our abilities to perform machine-learnt retrieval. We shall leverage the power of dense vector representations of text extensively in this thesis. We will use cross-lingual sentence retrieval to accelerate cross-lingual dataset creation. We shall take advantage of dense retrieval to learn retrieval end-to-end with downstream question answering tasks, and to build very low-latency ODQA models.

Generation Unsupervised pretraining enables powerful new generation capabilities. We shall employ these to generate questions for data augmentation and to build novel semi-structured KBs. We shall also build powerful, flexible generative ODQA systems, going beyond the simple span-extractive QA paradigm.

Low data regimes In the first half of this thesis, we shall use these techniques in order to develop methods that reduce the supervised dataset demands of QA models. We shall explore this idea in the context of unsupervised and few-shot RC, and evaluate how well RC models can transfer to test-time languages they have no supervised training data for, before demonstrating how retrieval-augmentation enables much stronger unsupervised ODQA systems.

Parametric vs Non-parametric knowledge As mentioned above, pretrained models capture knowledge in their parameters, which we refer to as *parametric knowledge*. This knowledge can be “retrieved” by prompting the model with an appropriate query, drawing parallels to explicit retrieval over a textual knowlegebase. As such, “retrieval-free” approaches to ODQA have been recently proposed that attempt to directly answer questions (Petroni et al., 2019; Radford et al., 2019; Roberts et al., 2020). We shall deeply explore the relationship between this new kind of parametric knowledge and more explicit *non-parametric* knowledge, such as that indexed and retrieved by more traditional Information Retrieval (IR) systems, which is a major theme running throughout the second two parts of this thesis.

1.2 Thesis Overview and Contributions

Directly after this introductory chapter, we shall proceed with a comprehensive background chapter. This chapter will describe the QA tasks we consider in detail, provide a brief the historical account of the field and review common approaches and practices in modern ODQA and RC. The remainder of this thesis is comprised of three parts, each consisting of two complementary chapters describing a specific research focus. Their contents will be briefly described below, along with a summary of the main contributions from each chapter. However, before detailing the specific contributions from each chapter, we shall briefly discuss how the thesis is structured in terms of narrative structure. In addition, Table 1.1 presents the big-picture flow of topics visually, indicating how the thesis narrative evolves as we focus on different key research aspects of Question Answering.

1.2.1 Big Picture Overview

In the previous section, we highlighted our key aims and themes, and in this section, we shall set out how the research in this thesis is structured, and how each topic leads on from the last. To recapitulate, our aims are to extend both the capabilities of QA models, and our understanding of them. We stated that models are 1) data hungry, 2) struggle to represent, store and use knowledge effectively 3) are often inflexible and complex pipelines, and 4) have slow, expensive inference. We shall tackle these four themes in turn, starting with data sample efficiency.

We will start with the RC task. The RC task is a basic, core competency of QA, and whilst it is a simpler setup that our ultimate goal of ODQA, it is important pre-requisite to understand it and perform it well. RC is therefore the focus of Part I, where we study in detail how to reduce the amount of annotated data needed for QA tasks. In Part II, we continue our attention on the low-to-no training data regimes set out in Part I, but move to the context of open-domain QA (which is the primary focus of this thesis), exploring how retrieval can be combined with pretrained models to perform open-domain QA-like behaviour. In doing so, we shall identify the concepts of parametric and non-parametric memory as a useful framework for discussing how models can represent, store and use knowledge. We shall develop these ideas throughout Part II, developing flexible models that can use hybrid knowledge access mechanisms. Finally Part III seeks to deepen our understanding of the ideas on parametric and non-parametric knowledge began in Part II, by studying whether knowledge required for QA tasks is present in the training data relative to a background corpus. These insights will allow us to develop models that are very fast and efficient at test time, which represent knowledge in a novel way.

Cutting through this linear narrative are a number of other recurrent aspects. QA is an empirically-driven field, and thus data and datasets play a very important role. Accordingly, we create, construct and analyse datasets closely in this thesis. In all cases, we use varying degrees of automation to reduce (or even completely remove in some cases) the amount of human annotation required to create datasets for QA. Chapter 4 takes a fairly traditional approach, where we create a dataset from scratch to measure a phenomenon of interest (multi-lingual transfer), but a multi-lingual retrieval/alignment tool is used to reduce the amount of manual translation required. Chapter 7 focuses on adding additional meta-annotations to existing, widely-used datasets, and uses another retrieval technique to find the most important instances that require annotation, reducing the human cognitive load needed for annotation. We shall also explore fully-automatically constructing datasets for QA, making use of automatic question generation. Chapter 3 will use this technique to induce RC models which need no human annotation. In chapter 8 – the last research chapter of the thesis – we return to this idea, in some ways coming full-circle in our research journey, albeit applying it in a different way, to create a new class of ODQA model, with a number of benefits over standard systems with respect to our stated aims above.

Now that the larger thesis structure and narrative flow has been laid out, we are in a position to describe the content of in each chapter in more detail. Thus, the following is a more detailed summary of each chapter, including its key research contributions.

	Part I		Part II		Part III	
	Chap 3	Chap 4	Chap 5	Chap 6	Chap 7	Chap 8
Reading Comprehension	●	→ ●	┄	┄	┄	○
Open-domain QA			●	─	→	●
Low data Regimes	●	→	●			
QA Dataset Construction	●	→ ●			●	→ ●
Generation	●			●	→	●
Parametric vs Non-parametric Mem.			●	─	→	●
Retrieval		●	─	→	─	●

Table 1.1: An overview of the tasks (top two rows), and key techniques/themes (bottom 5 rows) in this thesis, depicting how the chapters are connected and research ideas flow across the thesis. Black circles or lines indicate the topic is a key focus or heavily used in this chapter. A grey line indicates that whilst the technique is not the focus of the chapter, it still plays an important modelling role.

Part I: Reading Comprehension without Task Annotations

This part of the thesis concerns the task of span-based RC i.e. QA over a small textual knowledge source, on the order of 100 words in length, where answers are spans of text. There has been very strong empirical progress in recent years on this task, as measured by datasets such SQuAD (Rajpurkar et al., 2016). However, this has been predicated on access to a large, clean human-annotated dataset, greatly limiting the domains and languages for which such systems can be applied. In Part I we explore the extent to which zero-shot RC is possible – i.e. where no annotated RC data is available. We perform our study in two settings, described below.

Unsupervised Reading Comprehension by Cloze Translation First, in chapter 3, we develop and evaluate a method for training RC models when no RC training data is available *at all*. We develop a method which can generate a synthetic RC training dataset requiring no RC data, which we can then use to train off-the-shelf neural RC models. Our method exploits the relative ease in which *cloze questions* can be generated from text. This allows us to reduce the RC data generation problem to *cloze-to-natural question translation*, which we can tackle using advances in *unsupervised sequence translation*, driven by denoising pretraining. We demonstrate RC performance on SQuAD v1 at the level of early supervised approaches with access to 100,000 annotated data-points. Our main contributions are:

- This is the seminal work in the area of unsupervised RC.
- We demonstrate, thoroughly ablate and analyse a method for unsupervised RC based on generating a synthetic training set that can outperform some supervised models, without requiring billions of parameters.
- We provide an evaluation of few-shot RC performance, and demonstrate the effectiveness of synthetic data augmentation for this setting.

Evaluating Cross-Lingual Reading Comprehension Most languages do not have annotated RC data, which greatly complicates training RC models. One option is to apply the methods we describe in chapter 3 for the language of interest. However, this would not leverage annotated RC data that does exist in other, highly-resourced languages. In chapter 4, we consider the problem of zero-shot RC, assuming we have access to a large high-quality training dataset, albeit *in another language*. Whilst the lack of in-language training data is inconvenient, the lack of evaluation data is a more severe problem, critically limiting research and development in this important area. We therefore place a special emphasis on evaluation, carefully constructing a large, high-quality evaluation-only RC dataset, aligned across 7 languages. Using this dataset, we investigate the multilingual RC capabilities of a range of different multilingual approaches. Our main contributions are:

- We introduce a purpose-built, highly-parallel evaluation resource across 7 diverse languages for the task of zero-shot RC language transfer.
- We devise a novel annotation procedure, leveraging dense retrieval-based multilingual alignment, and the multilinguality of Wikipedia, allowing us to avoid extreme human translation workloads. This enables us to scale the size of the dataset, and allows for documents to be in their naturally-written language rather than manually translated.
- We formalize several cross-lingual QA tasks on the dataset, including a novel generalised cross-lingual task, and thoroughly evaluate a suite of models.

Part II: Retrieval-augmented Pretrained Models

Part I establishes our ability to tackle RC problems without relying on large amounts of annotated data. In the remainder of this thesis, we shift our attention to the more challenging and general *open-domain QA* task. Here, we will not be not provided with short paragraphs from which to read and produce an answer. Instead, Models must leverage knowledge distributed in a text corpus with millions or billions of words. Thus scalable techniques such as information retrieval are required to search for relevant knowledge. In Part II we shall apply retrieval-augmentation for ODQA and related tasks, and demonstrate its general utility for NLP tasks with demanding knowledge requirements. Part II is comprised of two chapters. In chapter 5, we stay on the theme of zero-shot QA established in Part I, this time in the context of the ODQA task. In Chapter 6, we shall then relax the low data requirement, and attack popular supervised ODQA tasks, and propose a class of end-to-end-trainable seq2seq retrieval-augmented model ideally suited for tasks such as ODQA.

How Context Affects Language Models' Factual Predictions We earlier introduced the concept of parametric knowledge. Parametric knowledge was demonstrated and quantified by Petroni et al. (2019) by querying a number of pretrained language models using cloze (fill-in-the-blank) questions requiring relational knowledge. It was found that pretrained models were able to answer a significant number of questions correctly, largely because they encode relational information in their parameters. This task, referred to as the LAMA probe, is equivalent to zero-shot ODQA with cloze questions. In chapter 5, we examine how the performance of these models vary when the cloze question is augmented with additional context. In doing so, we begin our investigation into how to complement and best draw out parametric knowledge, by using the mechanism of pre-pending questions with additional textual content. This chapters main contributions are:

- We show that, when supplied with an oracle context document, pretrained models dramatically improve on the LAMA probe, indicating that they can act as unsupervised RC models for cloze questions.
- We further demonstrate the promise of combining parametric and non-

parametric knowledge, by augmenting with context paragraphs retrieved from Wikipedia using a TF-IDF retriever. This is sufficient to improve BERT's performance on LAMA questions beyond early supervised retrieve-and-read ODQA models (Chen et al., 2017), despite being completely unsupervised.

Retrieval-Augmented Generation for Knowledge-Intensive NLP The above demonstrates the effectiveness of retrieval-augmentation formula for getting the benefits of both parametric and non-parametric knowledge. However, this is limited to answering cloze questions with single token answers, and it cannot easily leverage question-answer pairs available as training data. In chapter 6, we introduce a class of retrieval-augmented model capable of learning to generate free-form textual outputs, which we refer to as Retrieval-Augmented Generation (RAG). RAG models are comprised of a large, parametric-memory seq2seq generator, augmented with a non-parametric memory consisting of a dense vector index over Wikipedia. We formulate retrieved documents as latent variables, which can be marginalised out, enabling end-to-end fine-tuning without requiring document annotations. This means that RAG models inherit the immense flexibility of seq2seq pretrained language models, and can be fine-tuned solely on input-output pairs, such as question-answer pairs. Our main contributions are:

- We introduce the RAG model class, and demonstrate two RAG model formulations which can be optimised end-to-end to learn to retrieve relevant documents to augment the generator with.
- We validate RAG via experiments on ODQA tasks, but also demonstrate its flexibility by applying it on other challenging NLP tasks requiring knowledge.
- We analyse the interplay between parametric and non-parametric components via case studies and editing the non-parametric knowlegebase at test-time

Part III: Memorisation and Generalisation in ODQA

Part II explores how parametric and non-parametric knowledge can be combined via the mechanism of retrieval augmentation, and evaluates its effectiveness via open-domain QA. In Part III we shall remain in the area of open-domain QA, and concern ourselves with further developing our understanding of how models store, access and apply knowledge, specifically in the context of supervised ODQA tasks. Our focus will be *behavioural* in nature, and we shall deeply investigate the extent to which memorisation and generalisation are required for ODQA benchmarks.

Question And Answer Test-Train Overlap in Open-Domain QA Datasets In chapter 7, we set out a number of competencies that ODQA models should be able to exhibit in terms of difficulty and generalisation. By closely studying the test sets of popular ODQA tasks, we discover higher-than-expected numbers of questions that can be answered by various levels of memorisation of training time questions and answers. Using these findings, we evaluate a number of popular ODQA models to measure to what extent they actually generalise, and what drives their overall performance on ODQA benchmarks. We find that a recently-proposed class of parametric-only knowledge models, which are fine-tuned on the training set of question-answer pairs ("Closed-Book QA", Roberts et al., 2020), largely *only* memorise their fine-tuning data, and fail to meaningfully apply any knowledge from pretraining, especially for models smaller than 10B parameters. This effect is so strong that we can construct simple nearest neighbor models that compete with them. These models, which we call *"QA-pair retrievers"*, work by treating their training question-answer pairs (QA-pairs) as a non-parametric knowledgebase, and retrieve the most similar training question for a given test question, and return its answer. Our main contributions are:

- We set out a behavioural framework for ODQA generalisation based on memorisation of training data.
- We provide insights into how questions and answers are distributed between dataset splits, and to what extent each behaviour is required
- We evaluate a variety of models on these splits, and measure what kinds of QA behaviour different models achieve, demonstrating that simple nearest-neighbor models achieve results on par with closed-book QA models

65 Million Probably-asked Questions and What You Can Do With Them A key finding from the above is that treating QA-pairs as a non-parametric knowledgebase is competitive with parametric closed-book QA models. These QA-pair retrievers are also more memory-friendly than most approaches, with very fast inference, interpretable outputs (by inspecting retrieved QA-pairs), and the ability to easily update the model's knowledge at test time by adding or removing QA-pairs. However, the accuracy of both closed-book QA models and QA-pair retrievers lags far behind retrieval-augmented models like those developed in Part II. This is because the knowledge covered by training QA-pairs is a tiny fraction of that covered by the large textual knowledge sources used by retrieval-augmented models In chapter 8 we concern ourselves with how to improve models which memorise QA-pairs. Specifically, we want to understand how the accuracy of parametric and non-parametric models compare when the knowledge covered by their QA-pairs is not severely data-limited. To facilitate this, we construct a powerful automatic open-domain question-answer pair generator. We apply this generator at scale, generating a dataset of 65M QA-pairs from Wikipedia. We then develop strong QA-pair retrievers which use this dataset as a semi-structured knowledgebase. These models demonstrate accuracy on par with RAG, whilst being significantly faster at inference time. We also use the dataset to improve parametric-only models, but they still trail their non-parametric explicit retrieval analogues. Our contributions are:

- We introduce a novel generation pipeline for ODQA question generation, and use it to generate a very large dataset of QA-pairs
- We build stronger QA-pair retrievers which index this data and show that automatic generation and retrieval enables generalisation through memorisation
- We demonstrate the how the QA-pair retriever can be optimised for best-in-class trade offs between memory, speed, and accuracy
- We develop a method to combine QA-pair retrievers with slower, more general text-based retrieval-augmented models, demonstrating state of the art results and 2x latency improvements.

1.3 Open-sourced Materials

This thesis is accompanied by a large amount of open-sourced data, code and models. An exhaustive list of the open-sourced material supporting the work in this thesis, and where to access them, may be found in Appendix A.

Open-sourced code enables question generation and answering, multilingual QA evaluation, tools for running the LAMA evaluation, training and inference for RAG models, evaluation over diagnostic test QA splits, and code for running RePAQ models. Code is accompanied with open-sourced trained models enabling reproduction of key results.

All newly-created data resources in this thesis are also open-sourced. Open-sourced data includes a large dataset of automatically-generated QA data from chapter 3, the MLQA dataset created in chapter 4, the meta-annotations from chapter 7, and the PAQ dataset from chapter 8.

1.4 Published Material

This thesis is based on a number of previously published articles, which are listed here. Individual contributions not made the thesis author will be indicated at the beginning of relevant chapters. A list detailing open-sourced code, models and data can be found in Appendix A. Part I is based on the following papers:

- **Patrick Lewis**, Ludovic Denoyer, and Sebastian Riedel. 2019. Unsupervised Question Answering by Cloze Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*
- **Patrick Lewis**, Barlas Oğuz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. MLQA: Evaluating Cross-lingual Extractive Question Answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*

Material featuring in Part II first appeared in:

- Fabio Petroni, **Patrick Lewis**, Aleksandra Piktus, Tim Rockäschel, Yuxiang

Wu, Alexander H. Miller, and Sebastian Riedel. 2020. How Context Affects Language Models' Factual Predictions. In *Proceedings of the 2nd Annual Automated Knowledge Base Construction Conference (AKBC)*. Best Paper Award Recipient.

- **Patrick Lewis**, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*

as well as drawing on minor aspects of material from:

- Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, **Patrick Lewis**, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language Models as Knowledge Bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*
- Vladimir Karpukhin, Barlas Oğuz, Sewon Min, **Patrick Lewis**, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*
- Fabio Petroni, Aleksandra Piktus, Angela Fan, **Patrick Lewis**, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a Benchmark for Knowledge Intensive Language Tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*

Finally, Part III is based on the following publications:

- **Patrick Lewis**, Pontus Stenetorp, and Sebastian Riedel. 2021. Question and Answer Test-Train Overlap in Open-Domain Question Answering Datasets. In *Proceedings of the 16th Conference of the European Chapter of the Asso-*

ciation for Computational Linguistics (EACL). Best Paper Award Recipient.

- **Patrick Lewis**, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021. PAQ: 65 Million Probably-Asked Questions and What You Can Do With Them. *Transactions of the Association for Computational Linguistics (TACL)*

and includes minor material from:

- Sewon Min, Jordan Boyd-Graber, Chris Alberti, Danqi Chen, Eunsol Choi, Michael Collins, Kelvin Guu, Hannaneh Hajishirzi, Kenton Lee, Jennimaria Palomaki, Colin Raffel, Adam Roberts, Tom Kwiatkowski, **Patrick Lewis**, Yuxiang Wu, Heinrich Küttler, Linqing Liu, Pasquale Minervini, Pontus Stenetorp, Sebastian Riedel, Sohee Yang, Minjoon Seo, Gautier Izacard, Fabio Petroni, Lucas Hosseini, Nicola De Cao, Edouard Grave, Ikuya Yamada, Sonse Shimaoka, Masatoshi Suzuki, Shumpei Miyawaki, Shun Sato, Ryo Takahashi, Jun Suzuki, Martin Fajcik, Martin Docekal, Karel Ondrej, Pavel Smrz, Hao Cheng, Yelong Shen, Xiaodong Liu, Pengcheng He, Weizhu Chen, Jianfeng Gao, Barlas Oğuz, Xilun Chen, Vlad Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Wen-tau Yih. 2021. NeurIPS 2020 EfficientQA Competition: Systems, Analyses and Lessons Learned. *Proceedings of Machine Learning Research*

Chapter 2

Background

In this chapter, we shall introduce the field of Question Answering (QA) at a high level, including some basic scoping, highlighting of key characteristics, and a brief historical overview. We shall then cover the retrieval and generation techniques which we regularly employ in this thesis. Finally we will return to QA, describing in detail the two key tasks explored in this thesis, reading comprehension (RC) and open-domain QA (ODQA), including formalism, datasets, and evaluation.

2.1 Question Answering Overview

QA, as described in the literature, is a broad, multi-disciplinary and somewhat subjectively-delineated field, combining NLP, Information Retrieval, knowledge representation, machine learning and machine reasoning. Defining what is – and is not – QA is a non-trivial, and sometimes contentious topic. Some have argued that QA is sufficiently general in order to subsume all NLP tasks (Kumar et al., 2016; McCann et al., 2018). Others have suggested that the community conflates QA as a task, as distinct from QA as a modelling format (Gardner et al., 2019). We offer the following definition for the tasks which we refer to as QA in this thesis:

Question Answering is a family of tasks which require an answerer to produce concise natural language answers to questions, posed in natural language, by drawing on knowledge from a provided knowledge source.

Tasks which we consider to be QA in this thesis have questions which express a request for information from the supplied knowledge source, which must then be digested and summarised into an answer at an appropriate level of detail for the task at hand. We will use the symbol q to refer to questions, a to refer to answers and $\mathscr{C}$ to refer to the knowledge source which models draw upon to answer questions. We'll define questions $q \in Q$ and answers $a \in A$ to be random variables where Q and A are the sets of all possible questions and answers. We assume that there is a distribution of questions and answers, conditional on $\mathscr{C}$,

$$P(q, a|\mathscr{C})$$

and the high-level goal of QA is build models p that can model P, from which we can draw answers a^* to our questions

$$a^* = \arg\max_a p(a|q, \mathscr{C})$$

QA models p are typically parametrized, and machine learning techniques are employed to optimise the models parameters θ to minimise the empirical loss as defined by some loss function $\mathcal{L}$ over a dataset $\mathcal{D}$ of question answer pairs $\{(q_i, a_i)\}_{i=1}^{n}$ drawn from P. We shall return to formalism later in this chapter, after introducing some core conceptual and scoping material.

2.1.1 Knowledge Source, $\mathscr{C}$

The knowledge source does much to dictate the space of possible questions and answers. It acts as the representation of the world in which the QA system occupies. A system should not be expected to answer questions which require specific knowledge not present in their knowledge source. Moreover, valid answers must be consistent with the knowledge in the knowledge source. Knowledge sources vary in two main axes: Modality, and Scale:

Modality Our definition specified that questions and answers are expressed in natural language, but in general, knowledge takes a variety of forms. Knowledge-base

QA (KBQA) is term used when the knowledge takes the form of a structured knowledgebase. Visual QA (VQA, Antol et al., 2015, inter alia) tackles knowledge in the form of images or videos. The knowledge source can even take the form of a 3-dimensional environment in relation to a models' physical sensors in embodied QA (Das et al., 2018). The knowledge source could even be some multi-modal combination of all of the above (Oguz et al., 2020; Talmor et al., 2021). As mentioned in Chapter 1, we restrict ourselves to textual knowledge sources in this thesis, but many of the techniques that we develop are applicable to other modalities.

Scale The scale and scope of the knowledge source also has a profound effect on how we tackle QA tasks. For some tasks, the knowledge source is small and localised, such as a paragraph of text. These tasks are referred to as reading comprehension (RC) in the literature. On the other extreme, the knowledge source could be a whole book, encyclopedia, or even the entire internet, which is referred to as Open-domain QA (ODQA). ODQA can be regarded as a more general and challenging version of RC, since the modelling capabilities required to perform RC are still required in ODQA, along with the additional challenge of applying said behaviours over a much larger knowledge source.

Note that for some QA tasks, there may be no explicit mention of a knowledge source. This is usually either a notational convenience due to an assumption that the knowledge source is fixed, or in cases where we have extracted all necessary knowledge into some intermediate form which we do acknowledge, such as pretrained parameters. This is common in some modelling techniques in ODQA, such as closed-book QA (see section 2.4.2.2 later).

2.1.2 Answers, a

Our definition specified that answers should be expressed in natural language, but the level of abstraction and detail of answers can vary.

In some cases, the knowledge source can constrain the answer space, such as in span-extractive QA, where the answer must be a span of text from the knowledge source, or in KBQA, where the answer is typically a database entity. In other cases, the *task format* will restrict the answer space – for example, in multiple choice or yes/no QA, the output space is restricted to a handful of provided options.[1] In other cases, answers could be derived numerical quantities, like in some mathematical and logically-oriented tasks, such as DROP (Dua et al., 2019).

In the most general case, answers will be unconstrained natural language, and as such, the answer output space is combinatorially large. Modelling such tasks is a key concern of chapter 6. Answers can also differ in the level of detail or abstraction required. It is most common to only require short answers, such as noun phrases, dates, quantities or named entities. More challenging tasks, such as abstractive and long-form QA require the synthesis of sentential or paragraph-length answers, such as MSMARCO's NLG task (Bajaj et al., 2016) and ELi5 (Fan et al., 2019).

2.1.3 Questions, q

Questions are natural language statements intended to elicit information. Questions can be full of nuance, for example being ambiguous, noisy or under-specified, such as the question in Example A in Figure 1.1 from the introduction, or being inherently complex or multi-hop in nature (Bao et al., 2016; Talmor and Berant, 2018; Yang et al., 2018). In this thesis, we will encounter two main types of questions i) *natural questions*, which are the regular questions as one would use to ask to a question to a human, and, which in many European languages end with a question mark ii) *cloze questions* which are natural language statements with a blank which indicates where an answer should filled.

2.2 Historical Perspective

In this section we shall make a brief account of the history of the field, highlighting broad trends and recent changes. This section represents a personal perspective and reflection on the field, inspired by a diverse range of conversations and sources, as well as being influenced by Chen and Yih (2020) and Rodriguez and Boyd-Graber (2021). The notion of answering natural-language questions with computers ap-

[1] This does not imply that such tasks are always easier than other QA tasks (Clark et al., 2019)

pears around the time when computer science itself was first established as a concrete discipline. For example, the Turing test (Turing, 1950) posits that a conversation a computer and a human "Interogator" may be a practical way to measure machine intelligence.

By the mid-1960s, several attempts at building QA systems had been made (Simmons, 1965). These early efforts were characterised by heavy emphasis on syntax analysis of questions, driven by manually-written dictionary matching rules and heuristics. BASEBALL (Green et al., 1961), an early KBQA model, was able to perform a form of semantic parsing to map questions to database queries, to be executed on a database. Protosynthex (Simmons et al., 1964) and the Automatic Language Analyzer (ALA, Thorne, 1962) were some of the earliest attempts to tackle text-based ODQA resembling the kind we study in Parts II-III. Both extract symbolic meaning representations from their textual corpora (an encyclopedia, and a book on astronomy respectively) using rudimentary rules and heuristics to build an intermediate structured KB. Similar meaning representations are parsed from questions, before a simple matching of the symbolic question and knowledge representations is performed. This strategy would remain dominant for several decades.

Meanwhile, foundational work was being performed in IR, such as indexing documents by constituent words, measuring textual similarity by word overlap (Luhn, 1957), probabilistic relevance modelling (Maron and Kuhns, 1960), and the introduction of standard "Cranfield" evaluation protocols (Cleverdon, 1967).

The following decades would see progress in psycholinguistic theories of RC (Kintsch and van Dijk, 1978; Perfetti et al., 2001), and hypotheses of how to computationally tackle QA tasks (Schank and Abelson, 1977; Lehnert, 1977). This period would also see IR mature on smaller text corpora (∼100-1000 documents) with the development of term-weighted formulations (Singhal, 2001; Salton and Buckley, 1988; Robertson and Jones, 1976). However, actual end-to-end QA was generally slow to improve (Chen, 2018).

Empirical progress gathered pace in the 1990s with the introduction of community-wide datasets and shared evaluation procedures. In particular, the Text Retrieval Conference (TREC), first held in 1990, introduced a focus on large-scale retrieval, providing corpora and shared tasks to spur research (Robertson, 2008). TREC was (and still is) successful at accelerating information retrieval research, and is estimated to have resulted in search engine improvements that saved over 3 billion person-hours between 1999-2009 in the U.S.A. alone (Rowe et al., 2010).

Following successes in document retrieval, the beginning of the 2000s brought a renewed interest into end-to-end QA and RC. Hirschman et al. (1999) introduced DeepRead, an RC system and dataset, and the "Reading Comprehension Tests as Evaluation for Computer-Based Language Understanding Systems Workshop" at NAACL 2000 saw a number of other QA models introduced (Brill et al., 2000; Riloff and Thelen, 2000; Charniak et al., 2000; Wang et al., 2000). These systems, whilst more sophisticated and powerful than predecessors, still relied heavily on rule-based components, heuristics and shallow linguistic parsing.

Dedicated QA tracks were held at TREC between 1999-2007 (Voorhees and Harman, 1999; Voorhees and Tice, 2000). Modelling approaches at this time began to resemble what we today refer to as retrieve-and-read ODQA models (which we shall discuss in section 2.4.2.1), albeit in a much more modular, pipe-lined form. Typically, questions would be analysed to determine probable answer types (Li and Roth, 2002), and keyword-based search queries would be formulated, before executing IR over a corpus of (usually heavily-preprocessed) documents. An answer would be extracted from the search results, informed by the outcome of the answer type detection. These modules were comprised of complex ensembles of syntax parsers, semantic parsers, and rule-based components, with classifiers to combine and rank hypotheses (Chu-carroll et al., 2003; Moldovan et al., 2003).

Despite their complexity, and the labour required to construct them, these systems can be very effective. The most remarkable example is Watson DeepQA, which outperformed the best human *Jeopardy!* players (Ferrucci et al., 2010). This result garnered worldwide attention and greatly increased interest in QA (Chen and Yih, 2020). This period is also where span-extractive QA began to gain traction.

The later 2000s and early 2010s were typified by increasing popularity of machine learning techniques, in particular, the use of supervised machine learning coupled with manually-defined feature sets. Such techniques were powered by the introduction of new datasets, such as MCTEST (Richardson, 2013) for RC, and WebQustions (Berant et al., 2013) for ODQA.

Deep Learning and neural networks, which had been popular modelling approaches in the 1980s and 1990s, began to see a resurgence in the early 2010s , most famously in computer vision (Krizhevsky et al., 2012) but also in NLP (e.g. Collobert et al., 2011). Neural representation learning methods such as Word2Vec (Mikolov et al., 2013) were particularly influential. Deep learning techniques were especially attractive for learning in an end-to-end manner from unprocessed inputs. As a result, much of the complex modular pipeline of previous approaches could be replaced by a single neural model, reducing complexity and the risk of cascading errors.

However, supervised deep learning models were more data hungry than their manually-featurised SVM and logistic predecessors, and larger datasets would be required. Hermann et al. (2015) introduced the CNN/Daily Mail RC dataset, accompanied by one of the first neural RC models. Hill et al. (2016) constructed the Children's Book Test, and applied a neural memory network on it. In particular, the SQuAD (Rajpurkar et al., 2016) dataset captured the community's imagination (see section 2.5.2.2 later). The combination of clean data, simple evaluation, and leaderboard in SQuAD (and other datasets of its ilk e.g. MSMARCO (Bajaj et al., 2016) and TriviaQA (Joshi et al., 2018)) spurred research in RC (Wang and Jiang, 2017; Seo et al., 2017; Weissenborn et al., 2017b; Wang et al., 2017; Clark and Gardner, 2018; Yu et al., 2018). There was rapid empirical progress, with models reaching supposed average human test-set accuracy in early 2018 (Linn, 2018). This period was typified by custom deep learning architectures, designed specifically for the task at hand, incorporating special attention mechanisms and gated RNNs (Hochreiter and Schmidhuber, 1997; Cho et al., 2014; Bahdanau et al., 2015).

This period also saw deep-learning approaches to IR – particularly supervised and semi-supervised approaches – begin to gain traction. Early approaches, such as S2NET (Yih et al., 2011) and DSSM (Huang et al., 2013), demonstrated the use of neural retrievers trained on large amounts of click-through data. Models combining term-matching and neural dense representations gained popularity in the mid 2010s, and still dominate today (Guo et al., 2016; Mitra et al., 2017, inter alia).

The mid-2010s to the present has witnessed a proliferation of datasets, examining and isolating a vast range of different QA phenomena. Indeed, the growth in annotated resources for QA in English is one of the hallmarks of the modern QA research landscape, to such an extent that it is becoming impractical for individual researchers to keep track (Rogers et al., 2021; Cambazoglu et al., 2021). Perhaps as a result, there have been recent efforts to aggregate datasets together into benchmark collections that simultaneously test many skills at once. This first became prevalent in Natural Language Inference, such as GLUE (Wang et al., 2018a), but has also been applied to QA and IR (Fisch et al., 2019; Thakur et al., 2021). In Section 2.5 we shall discuss in more detail the most relevant datasets to this thesis.

A significant development in ODQA came with the popularisation of combining passage retrieval from Wikipedia, with neural RC models, most famously by Chen et al. (2017) and Clark and Gardner (2018). This two-stage retrieve-and-read paradigm (see section 2.4.2.1) is much simpler than the complex pipelines of systems like the *Jeopardy!*-winning DeepQA, and has proven to be empirically very strong, representing the state-of-the-art approach at time of writing.

The last five or so years have continued to see rapid progress in QA, and in machine learning and NLP in general, powered by more data, aggressively increased compute (Hernandez and Brown, 2020), improved tools and software for machine learning (Al-Rfou et al., 2016; Martín Abadi et al., 2015; Paszke et al., 2019), and a large influx of personnel (Zhang et al., 2021). We finish our historical overview by discussing the following trends we have observed in the last 5 years:

Unsupervised Pretraining The first recent trend is the maturation of unsupervised representation learning in NLP. This body of work seeks to train high-quality general-purpose representations from text, which can then be used downstream. These ideas were first embraced in the form of initialising word embedding matrices in neural models using pretrained word embeddings (Mikolov et al., 2013; Pennington et al., 2014) in order to reduce cold-start and out-of-vocabulary problems. Methods such as CoVE (McCann et al., 2017) and ELMo (Peters et al., 2018) extended this idea by initialising models with pretrained contextual encoders. The final step in this evolution was to shift to an entirely-pretrained model, typically a large, general purpose-model like a Transformer (Vaswani et al., 2017), pretrained on a language modelling or denoising task, which can then be fine-tuned via back-propagation on the downstream task at hand. Large Transformer-based pretrained models such as GPT (Radford et al., 2018) and BERT (Devlin et al., 2019) quickly established themselves, and the field has experienced a period of consolidation. The latest generation of generative transformers, such as BART (Lewis et al., 2020a), T5 (Raffel et al., 2020) and GPT2/3 (Radford et al., 2019; Brown et al., 2020) have become the default method across essentially all NLP tasks.

Increasing emphasis on behavioural testing and analysis Whilst progress on in-domain test sets improved, it became increasingly clear that the combination of large crowd-sourced datasets and neural models are no panacea. One downside of deep neural models, especially pretrained ones, is their lack of interpretability. A large body of research has demonstrated that such models may give strong results on static, I.I.D. test sets, but may suffer in out-of-distribution testing, such as under adversarial testing or under domain change (Sugawara et al., 2018; Jia and Liang, 2017; Min et al., 2018; Talmor and Berant, 2019; Bartolo et al., 2020). Recently, state-of-the-art RC models have demonstrated robustness to some of these effects (Bowman, 2021), but domain transfer remains challenging. Moreover, these models' performance may drop when restricting the amount of training data (Yogatama et al., 2019). Techniques such as human-and-model-in-the-loop data creation (Wallace et al., 2019; Bartolo et al., 2020; Kiela et al., 2021), behavioural testing (Ribeiro et al., 2020) and behavioural evaluation sets such as those in chapter 7 have been introduced to better understand, diagnose and hopefully mitigate pathologies in NLP models. Such models also tend to express the underlying biases in their training data (Zhao et al., 2017; Dinan et al., 2020, inter alia.). Given the wide industry adoption of such models, *Responsible AI* research – which combines ethics, sociology and AI – looks at understanding and mitigating models' harmful expressions of bias, has become an area of vital importance (Dignum, 2019).

Scale – in all its aspects Pretrained models appear to improve dramatically by scaling data, model size and compute. State-of-the-art models have grown from a few million parameters in 2018 to billions (Radford et al., 2019), tens of billions (Raffel et al., 2020), hundreds of billions (Brown et al., 2020) and even trillions of parameters (Fedus et al., 2021) at time of writing. These models are trained on ever-larger corpora, with recent pretraining corpora weighing in at 800GB of raw text (Gao et al., 2021a). All of this requires extreme resources, putting pretraining beyond the reach of most researchers. Billion-parameter-plus models can exhibit qualitatively-different emergent behaviour to otherwise-equivalent smaller models, such as few-shot "in-context" learning (Brown et al., 2020). Whilst we likely have yet to see performance gains from scaling model and data sizes saturate, concerns about equitable access to such models, and environment impact are growing (Strubell et al., 2019). As such, an "Anti-scale" line of research has begun to develop, producing modestly-sized models that can compete with vast ones (Schick and Schütze, 2021a, inter alia.). Researchers have also sought to scale the amount of context a model can consume, usually by proposing modifications to self-attention, so far without convincing consistent gains over standard transformers (Tay et al., 2020, 2021). In fact, architectures developed for ODQA, such as those we consider in Parts II-III represent some of the best available options for models which can take entire web-scale corpora as input at inference time.

2.3 Retrieval and Generation Modelling Techniques

In this section, we shall briefly cover relevant background material on the information retrieval (IR) and natural language generation (NLG) techniques which we heavily leverage in this thesis.

2.3.1 Natural Language Generation

We exploit NLG to automatically-generate questions in order to build QA models (chapters 3 and 8) or employ NLG architectures to improve the applicability, quality and flexibility of QA systems (chapter 6). For more comprehensive details on NLG, the reader is referred to Reiter and Dale (2000) and Gatt and Krahmer (2018), and to Ji et al. (2020) for an overview of state-of-the-art practices.

2.3.1.1 Cloze Question Generation and Rule-based Systems

In general, we will encounter the need to generate questions from answers a embedded in textual context c. Consider the (c,a) pair "The London Sevens was the last tournament of each season but the Paris Sevens became the last stop on the calendar in 2018." In the special case that we require cloze questions rather than natural questions, question generation is straightforward. A simplification step can be carried out, e.g. by performing a syntax parse and keeping the answer-containing sub-clause, to yield: "the Paris Sevens became the last stop on the calendar in 2018". A cloze question can then be trivially obtained by replacing the answer with a blank: the Paris Sevens became the last stop on the calendar in ___. The relative ease of cloze question generation will be exploited in chapter 3 as an intermediate step to tackle the more difficult natural question generation problem.

Sophisticated rule-based systems were a popular statement-to-natural-question method until relatively recently. We shall encounter such a method from Heilman and Smith (2010) in chapter 3. This method works by defining a pipeline of syntactic transformations on (c,a) pairs, namely, simplification (as above), main verb decomposition, and finally Wh* phrase insertion and movement.[2] There are usually several rules per step, leading to many candidate outputs, which are then ranked by a

[2] We use the popular term "Wh* word/phrase" to refer to interrogative words/phrases.

machine-learnt ranker. Such systems rely on a large number of rules, a high-quality syntax parse, and a supervised ranker, which can make them brittle.

2.3.1.2 Sequence-to-Sequence Models

We will encounter a number of cases where we need to generate a sequence of tokens, conditioned some other input sequence. This class of problem can be tackled by sequence-to-sequence (seq2seq) learning (Sutskever et al., 2014; Cho et al., 2014; Bahdanau et al., 2015). Seq2seq models often follow an *encoder-decoder* architecture, whereby the encoder is a neural network that encodes the input sequence, and the decoder, conditioned on the encoded state, auto-regressively generates the output token-by-token. The encoder and decoder are usually recurrent neural networks such as LSTMs or GRUs (Hochreiter and Schmidhuber, 1997; Cho et al., 2014), or, more recently transformer encoder-decoders (Vaswani et al., 2017).

Our goal is to train a model $p_{s\to t}(y|x)$ for source sequences $x = (x_1, \dots, x_n) \in \mathcal{S}$ and target sequences $y = (y_1, \dots, y_m) \in \mathcal{T}$ and all tokens $x_k, y_k \in \mathcal{V}$, where $\mathcal{S}$ and $\mathcal{T}$ are the space of source and target sequences respectively, and $\mathcal{V}$ is the vocabulary. Our model will be composed of a seq2seq encoder-decoder

$$p_{s\to t,\theta} = \mathsf{DEC_t}\left(\mathsf{ENC_s}(\cdot)\right)$$

We will assume access to a dataset of aligned pairs $\mathcal{D}_{s,t} = \{(\mathsf{x}_i, \mathsf{y}_i)\}_{i=1}^{N}$, with which to train a model. We define our model as follows:

$$p_{s\to t}(y|x) = p_{s\to t}(y_1, \dots, y_m|x) = \prod_{j=1}^{m} p_{s\to t,\theta}(y_j|y_1, \dots, y_{j-1}, x)$$

To train, we minimise negative log-likelihood on $\mathcal{D}$ via stochastic gradient descent

$$\mathcal{L}_\theta = -\sum_{i,j}^{N,m} \log p_{s\to t,\theta}(y_{i,j} = \mathsf{y}_{i,j}|\mathsf{y}_1, \dots, \mathsf{y}_{j-1}, \mathsf{x})$$

Inference involves searching for the maximum likelihood target $\arg\max_{y\in\mathcal{T}} p_{s\to t}(y|x)$ for which effective approximations such as beam search are available.

In the latter chapters of this thesis, we shall make extensive use of generative seq2seq pretrained model such as BART (Lewis et al., 2020a) and T5 (Raffel et al., 2020). These models are pretrained to generate clean outputs given masked or noised inputs, acting as seq2seq denoising auto-encoders (Vincent et al., 2008). In the formalism above, we arrive at such models by simply defining $x = \mathsf{MASK}(y)$, where MASK is a noising function[3] and using a very large dataset $\{(\mathsf{MASK}(\mathsf{y}_i), \mathsf{y}_i)\}_{i=1}^{N}$. Such models are extremely flexible, providing the benefits of the pretrained contextual representations from BERT with the added ability of being able to generate high quality text sequences.

2.3.1.3 Back-Translation, Cycle-consistency, Unsupervised MT

In general, if we can train a model $p_{s\rightarrow t}(y|x)$ which translates from the source to target domain, we can also train a backwards model $p_{t\rightarrow s}(x|y)$ using the same dataset. We often encounter the case where we have access to only a limited dataset of aligned pairs $\mathcal{D}_{s,t} = \{(\mathsf{x}_i, \mathsf{y}_i)\}_{i=1}^{N}$, but large amounts of unpaired data from each domain, $\mathcal{D}_s = \{\mathsf{x}_l\}_{l=1}^{L}$ and $\mathcal{D}_t = \{\mathsf{y}_k\}_{k=1}^{K}$, where $L, K >> N$. For example, much more unpaired English and French text exists than English-French parallel pairs.

A technique called back-translation (Sennrich et al., 2016) uses the backwards model $p_{t\rightarrow s}$ to produce pseudo-translations $u(y) = \arg\max_x p_{t\rightarrow s}(x|y)$, enabling us to transform $\mathcal{D}_t$ into a large pseudo-parallel training set $\hat{\mathcal{D}}_t = \{(u(\mathsf{y}_k), \mathsf{y}_k)\}_{k=1}^{K}$ which can serve as data-augmentation for the forward model $p_{s\rightarrow t}$. An analogous procedure employing $v(x) = \arg\max_y p_{s\rightarrow t}(y|x)$, could be applied to generate training data for the backwards model. This technique takes advantage of the *cycle consistency* principle, namely, $u(v(x))$ should equal x and $v(u(y))$ should equal y. Cycle consistency provides supervision signals without requiring paired data.

It turns out that we can learn $p_{s\rightarrow t}$ and $p_{t\rightarrow s}$ without requiring *any* paired $\mathcal{D}_{s,t}$, only a large amount of unpaired data $\mathcal{D}_s$ and $\mathcal{D}_t$. This trick works by combining parameter sharing, and the denoising language modelling and back-translation techniques mentioned above, and is referred to as Unsupervised Machine Translation (UMT,

[3] most often token-masking, but other noise functions have been used (Lewis et al., 2020a)

Lample et al., 2018c,a; Conneau and Lample, 2019; Artetxe et al., 2018). Here we train four models, comprised of shared encoders and decoders, to perform denoising and translation for the source and target domains respectively:

$$\begin{aligned} \text{Denoising models:} \quad & p_{t\to t} = \mathsf{DEC_t}\left(\mathsf{ENC_t}(\cdot)\right) \quad p_{s\to s} = \mathsf{DEC_s}\left(\mathsf{ENC_s}(\cdot)\right) \\ \text{Translation models:} \quad & p_{s\to t} = \mathsf{DEC_t}\left(\mathsf{ENC_s}(\cdot)\right) \quad p_{t\to s} = \mathsf{DEC_s}\left(\mathsf{ENC_t}(\cdot)\right) \end{aligned}$$

We start by training the denoising models $p_{t\to t}$ and $p_{s\to s}$ with objectives:

$$\mathcal{L}_{t\to t} = -\log p_{t\to t}(y|\mathsf{MASK}(y)) \qquad \mathcal{L}_{s\to s} = -\log p_{s\to s}(x|\mathsf{MASK}(x))$$

This enables the encoders $\mathsf{ENC_s}$ and $\mathsf{ENC_t}$ to learn how to encode (noisy) inputs, and the decoders $\mathsf{DEC_s}$ and $\mathsf{DEC_t}$ to generate clean outputs. After some training, we take the encoder for one domain, and apply the decoder for the other domain to create the translators $p_{t\to s}$ and $p_{s\to t}$. These models will be weak, but are capable of taking text in one domain and generating a noisy output in the other domain. We can improve the translators using cycle-consistency with back-translation losses,

$$\begin{aligned} \mathcal{L}_{s\to t} &= -\log p_{s\to t}(y|u(y)) \quad & \mathcal{L}_{t\to s} &= -\log p_{t\to s}(x|v(x)) \\ u(y) &= \arg\max_{x} p_{t\to s}(x|y) \quad & v(x) &= \arg\max_{y} p_{t\to s}(y|x) \end{aligned}$$

We continue to train both denoising and backtranslation losses,

$$\mathcal{L} = \alpha\left(\mathcal{L}_{t\to t} + \mathcal{L}_{s\to s}\right) + (1-\alpha)\left(\mathcal{L}_{s\to t} + \mathcal{L}_{t\to s}\right)$$

gradually decreasing α from $1 \to 0$ as the translators improve. Model selection is on based on a small parallel dataset, or using back-translated BLEU scores. For datasets where there is little parallel data, such as low resource machine translation, this approach outperforms alternative approaches (Lample et al., 2018a), and can be leveraged for general sequence transduction tasks, for example for question rewriting and style transfer (Subramanian et al., 2018; Perez et al., 2020). We shall use this technique for cloze-to-natural question translation in chapter 3.

2.3.2 Information Retrieval

IR deals with the task of finding the most relevant document c in a large knowledge source $\mathscr{C}$, given a question q. In QA settings, relevant documents are those that allow us to infer answer a, usually because the document contains the answer.

2.3.2.1 Sparse Term-based Retrieval

Supervised Dense retrieval systems have recently begun to outperform sparse term-based systems. Nevertheless, these retrievers are attractive due to their well-understood properties. They are relatively space-efficient and can be made extremely fast using inverted indices (Zobel et al., 1998). They also perform well in zero-shot settings (Thakur et al., 2021).

These methods define a vector space model, where documents c in the knowledge corpus and queries q are represented by a vector, $\mathbf{v}$ of *terms* (Salton et al., 1975). These terms are usually words, or n-grams, with each dimension of the vector space encoding a single term. Since there are many terms, and most terms do not appear in most documents, the vectors are sparse and high-dimensional. A *term-weighting* scheme is often used, generally revolving around "term frequency-Inverse document frequency" (TF-IDF), where for term i occurring in document c

$$v_{i,c} = \mathrm{tf}_{i,c} \cdot \log \frac{N}{\mathrm{df}_{i,}}$$

where $\mathrm{tf}_{i,c}$ is a function measuring the frequency of term i in c, and df_i measures the background frequency of term i. A high tf indicates the term is characteristic of the document, whereas a high document frequency indicates the word appears in many documents and carries little information. There are many options for specific weighting functions. The most enduring are inspired by probabilistic retrieval formulations (Robertson, 1977), such as BM25 (Robertson and Zaragoza, 2009). Retrieval involves calculating a relevance score between the query vector v_q and all document vectors v_c, usually a variant on dot product,

$$c^* = \arg\max_{c \in \mathscr{C}} \mathbf{v}_q^\top \mathbf{v}_c$$

Extensions to these models include query reformulation and pseudo-relevance feedback techniques, which modify the question's query vector.

2.3.2.2 Dense Retrieval

We refer to representing documents and queries using dense, continuously-valued embedding vectors, rather than sparse term-based vectors as above, as *dense retrieval*. Unsupervised dense embedding approaches, such as LSI (Deerwester et al., 1990), and supervised dense retrievers have been studied for decades (Yih et al., 2011). However, results were inconsistent, and required very large training datasets. Pretrained language models have improved the feasibility this approach, and supervised dense embedding models now empirically outperform most sparse approaches where 1000+ training pairs are available (Karpukhin et al., 2020, inter alia.). Dense embeddings are amenable for use in deep learning, and off-the-shelf toolkits for efficient, in-memory Nearest-Neighbour and Maximum Inner Product search, such as FAISS (Johnson et al., 2019) enable extremely fast and scalable retrieval.

A typical contrastive dense retrieval set up is briefly outlined below, based on DPR (Karpukhin et al., 2020). We again consider the problem of retrieving passages c from a corpus $\mathscr{C}$ of N passages, for questions q. We assume access to a set of pairs of m gold question-document pairs: $\{(q_i, c_i^+)\}_{i=1}^m$ which we use to train a supervised retriever. We obtain negative passages p^- by sampling from $\mathscr{C}$, perhaps using some importance sampling scheme. We thus have a training dataset:

$$\mathcal{D} = \{(q_i, c_i^+, c_{i,1}^-, \dots, c_{i,n}^-)\}_{i=1}^m$$

We define a learnt relevance score, based on a bi-encoder architecture[4]

$$\mathrm{sim}(q,c) = E_Q(q)^\top E_C(c) = \mathbf{v}_q^\top \mathbf{v}_c \qquad \mathbf{v}_q = E_Q(q) \qquad \mathbf{v}_c = E_C(c)$$

where E_Q and E_C are (usually BERT-based) encoders, and $\mathbf{v}_q$ and $\mathbf{v}_c$ are dense vector embeddings of questions and documents respectively. We train by maximising the score of gold passages relative to negatives, e.g. by minimising negative log likelihood of gold passages,

$$\mathcal{L}\left(q_i, c_i^+, c_{i,1}^-, \ldots, c_{i,n}^-\right) = -\log\left(\frac{e^{\mathrm{sim}(q_i, c_i^+)}}{e^{\mathrm{sim}(q_i, c_i^+)} + \sum_{j=1}^{n} e^{\mathrm{sim}(q_i, c_{i,j}^-)}}\right)$$

At test-time, we pre-compute document embeddings for all documents in the corpus. Retrieval requires solving the maximum inner product search (MIPS) problem below, approximately soluble in logarithmic time (Johnson et al., 2019)

$$c^* = \arg\max_{c \in \mathscr{C}} \mathbf{v}_q^\top \mathbf{v}_c$$

2.3.2.3 Rerankers and the Decomposability Gap

Scalable retrieval generally limits the choice of relevance function to be decomposable, enabling pre-computation and caching of document representations. Most practical choices of decomposable functions are some variant of inner product or L2 distance.[5] This requirement limits the expressibility of the relevance function, relative to *cross-encoders*, which use multiple layers of cross-attention between the document and question. Closing this *decomposability gap* (Seo et al., 2019) is an active area of research (Humeau et al., 2020; Khattab and Zaharia, 2020; Khattab et al., 2021) but cross-encoders are generally significantly stronger, despite their impracticality for large-scale ranking. A common compromise is to use a cross-encoder *reranker* after a decomposable retriever to re-score the top-k retrieved items.

[4] This architecture is often referred to as a Siamese network (Bromley et al., 1993), (or two-tower network), but we prefer the more inclusive bi-encoder terminology

[5] Transformations exist to map inner products to L2 distances, and vice versa, and other choices such as Mahalanobis and cosine distances (Mussmann and Ermon, 2016; Ram and Gray, 2012)

Rerankers are usually trained similarly to the dense retriever above, except that i) the negative passages will be sampled from the top-K passages from the decomposable retriever and ii) the relevance function will be non-decomposable, $sim(q,c) = E_{QC}(q,c)$, such as BERT taking the concatenation of the document and question as input. Whilst this approach almost always improves test-time relevance, it comes with at the expense of latency and computational cost. We employ such a strategy, and examine its tradeoffs in chapter 8.

2.3.2.4 Parametric Memory and "Generation as Retrieval"

As mentioned in chapter 1, the introduction of pretrained models has seen a new retrieval paradigm emerge. These models exhibit linguistic knowledge, e.g. structures resembling parse trees (Peters et al., 2018; Tenney et al., 2019b; Goldberg, 2019), but their training objectives also encourage learning *relational* knowledge.

For example, a model may be trained on the sentence "Dante was born in Florence". A masked language model (Devlin et al., 2019) may transform this into the training instance "Dante was born in [MASK]" with the task being to fill the mask token with "Florence". This training instance is equivalent to a cloze question, and such models could be viewed as cloze QA models. The training objective encourages storing relational knowledge in parameters, such as the relational triple (DANTE, born_in, FLORENCE), in order to successfully predict such instances.

This phenomenon was demonstrated and quantified by Petroni et al. (2019) and concurrently by Radford et al. (2019). Knowledge can be "retrieved" from a model by providing a *prompt*, such as a templated cloze question for a given relation. For example, for the relation born-in we define the prompt template "[SUBJECT] was born in the city of [OBJECT]". If we wish to retrieve the birth city of, e.g. Barack Obama, we can pass into the model "Barack Obama was born in the city of ___" and observe the model's prediction, in an analogous way to a relational KB, where we would query for triple (BARACK_OBAMA, born_in, ___).

Petroni et al. (2019) report that on a collection of simple relational facts mentioned in Wikipedia, using BERT to "retrieve" object entities was comparable in accuracy to an querying a structured KB constructed using a task-specific supervised relation-extraction system. Radford et al. (2019) go a step further, and directly provide the model with a natural (rather than cloze) question as a prompt, and achieving a non-trivial (but very low) 4% accuracy on NaturalQuestions.

This is a particularly intense area of research, and there is much we have yet to understand about the properties of knowledge stored in large language models. Relational knowledge seems to be a key area where scaling model size is effective (Kirstain et al., 2021). GPT3 (Brown et al., 2020) improves GPT2's score on NaturalQuestions to 14.5%, by increasing the parameter count by 116×. Prompting techniques have also been subject to intense study, with a great deal of recent work on discovering automatic prompts via search (Shin et al., 2020; Jiang et al., 2020a), and generalizing prompts as continuous conditioning states (Liu et al., 2021c; Zhong et al., 2021; Qin and Eisner, 2021; Lester et al., 2021; Li and Liang, 2021; Logan IV et al., 2021) In this thesis, we refer to the relational knowledge stored in pretrained model parameters as *parametric knowledge*. An ODQA approach based on parametric knowledge, *closed-book QA*, features heavily in Parts II-III and is described in detail in section 2.4.2.

2.4 Question Answering Modelling

In this section we will outline the current popular modelling approaches for RC and ODQA in more detail, as background material to contextualise and support the content that appears in following chapters.

2.4.1 Reading Comprehension

RC is a QA task where the knowledge source $\mathscr{C}$ consists of a single, short document, usually referred as the *context* or *passage*. Let c refer to the context, where $c = \mathscr{C}$ for later ease of notation. Typically, RC tasks (e.g. SQuAD) consist of extractive span-based QA, where an answer is defined as span of text within the context.

We assume access to a dataset $\mathcal{D} = \{(c_i, q_i, a_i)\}_{i=1}^{m}$ of m context, question, answer triples, where each context c is comprised of a sequence of l_c tokens, such that $c = (c_1, \dots, c_{l_c})$. Answers are comprised of a pair of indices $a = (a_{st}, a_{en})$ into c, such that a lexicalized answer is given by $(c_{a_{st}}, \dots, c_{a_{en}})$. Our goal is to learn a model $p\left(a|q,c\right) = p\left(\left[a_{st}, a_{en}\right]|q,c\right)$ such that the gold answer span a^* is given by

$$a^* = (a^*_{st}, a^*_{en}) = \underset{0<a_{st}<a_{en}<l_c}{\arg\max}\ p\left(\left[a_{st}, a_{en}\right]|q,c\right)$$

Directly modelling the joint of a_{st} and a_{en} requires modelling an output answer space which is quadratic in context length l_c. This can be difficult to learn, so it is common to assume the a_{st} and a_{en} are conditionally independent, and thus $p(a|q,c)$ be factorized to $p_{st}(a_{st}|q,c) \cdot p_{en}(a_{en}|q,c)$.[6]

Many specialised RC architectures exist, such as BiDAF (Seo et al., 2017), now largely superseded BERT-style fine-tuning. Generally, we pass in q and c into an encoder E, consisting of a number of cross-attention layers, to yield contextualised representations of the context tokens

$$\mathbf{H} = E(q,c) \qquad \text{where } \mathbf{H} = (\mathbf{h}_1, \dots, \mathbf{h}_{l_c})^\top,\ \ \mathbf{H} \in \mathbb{R}^{d \times l_c}$$

and p_{st} and p_{en} are defined as $\mathsf{softmax}(\mathbf{w}_{st}^\top \mathbf{H})$ and $\mathsf{softmax}(\mathbf{w}_{en}^\top \mathbf{H})$ respectively for trainable weight vectors $\mathbf{w}_{st}, \mathbf{w}_{en} \in \mathbb{R}^d$. Such factorised models are trained by minimising the following loss for gold answer spans $(\mathsf{a}_{\mathsf{st}}, \mathsf{a}_{\mathsf{en}})$

$$\mathcal{L}(\mathsf{a}_{\mathsf{st}}, \mathsf{a}_{\mathsf{en}}, q, c) = -\log p_{st}(a_{st} = \mathsf{a}_{\mathsf{st}}|q,c) - \log p_{en}(a_{en} = \mathsf{a}_{\mathsf{en}}|q,c)$$

Finally, RC can be approached as a purely seq2seq task, by learning to generate the answer text directly. This approach, does not take into account the inductive bias that the answer must be a strict substring from the context. Nevertheless, it can be competitive when using sufficiently-powerful pretrained seq2seq models.

[6] This assumption may result in a drop in modelling quality. Such as case will occur in chapter 8

2.4.2 Open-Domain QA

In the typical ODQA setting, the knowledge source will be comprised of text up to billions of times larger than the knowledge source in RC.[7]

Formally, the ODQA task is defined as follows. We assume access to a knowledge source $\mathscr{C}$ which we will further assume is segmented of a large number N of machine-readable passages $\mathscr{C} = \{c_n\}_{n=1}^{N}$. In the general case, we will not assume any structure to $\mathscr{C}$ other than that it is comprised of a set of passages c.[8] In order to learn a model, we assume access to a dataset of question-answer pairs, $\mathcal{D} = \{(q_i, a_i)\}_{i=1}^{m}$. In addition, it is common to have access to *passage-grounded answer annotations*, $\mathcal{D} = \{(\mathcal{C}_i, q_i, a_i)\}_{i=1}^{m}$ where $\mathcal{C} = \{c_1, \ldots, c_l\} \subset \mathscr{C}$, a small set of one or more passages which contains sufficient information to answer q.[9]

Some datasets do not provide answers grounded to specific passages, however, distant supervision techniques (Mintz et al., 2009) can be used to obtain approximate groundings. This is straightforward when answers can be assumed to be sub-strings of passages, but fuzzy matching and other techniques are available for other settings. For distantly-supervised grounding, marginalisation or EM methods may be used in training to mitigate noise in the grounding process (Min et al., 2019a).

2.4.2.1 Retrieve-and-Read Open-Domain QA

The most popular, and currently most accurate, approach to ODQA is referred to as "retrieve-and-read". This approach amounts to attempting to apply an RC model to the entire knowledge source (Chen et al., 2017; Choi et al., 2017; Clark and Gardner, 2018; Wang et al., 2018d; Lee et al., 2018a; Wang et al., 2018c; Lin et al., 2018; Min et al., 2018; Lee et al., 2018a, 2019a; Das et al., 2019; Xiong et al., 2019; Pang et al., 2019, inter alia.). However, since it is impractical to directly apply a reader to the whole knowledge source, a subset of promising passages are selected

[7] In-between the two in scale lies the task of document QA (Clark and Gardner, 2018), which we consider to be a special-case of ODQA with a small knowledge source

[8] Approaches that leverage additional structure, such as hyperlink graphs do exist, but are dataset specific, limiting their applicability (De Cao et al., 2019; Min et al., 2019b)

[9] More than one passage in $\mathcal{C}$ allows for information redundancy in the knowledge source, which can make some questions significantly easier to answer, and is relied upon for a technique referred to as *macro reading* (Mitchell et al., 2009)

for reading using information retrieval techniques.

The retriever component p_{ret} could be an untrained system such as BM25, or we can train e.g. a dense retriever (section 2.3.2.2). Retrieval training data can be obtained by re-purposing an ODQA dataset with passage-grounded answer annotations, by transforming $\mathcal{D}_{\text{odqa}} = \{(\mathcal{C}_i, q_i, a_i)\}_{i=1}^{m}$ into $\mathcal{D}_{\text{ret}} = \{(q_i, c_i^+)\}_{i=1}^{m'}$ by ignoring answers, and creating positive question-document pairs for each grounded answer passage $c^+ \in \mathcal{C}$. The reader component, typically an RC system of the kind described in section 2.4.1 can also be trained using a similar transformation on $\mathcal{D}_{\text{odqa}}$ into $\mathcal{D}_{\text{read}} = \{(q_i, c_i^+, a_i)\}_{i=1}^{m'}$, yielding $p_{\text{read}}(a|q,c)$

Inference in such systems usually follows a pipeline, where first, the top-k passages will be retrieved from p_{ret}, which are fed into p_{read} to produce an answer candidate from each passage. To produce a final answer, an aggregation step across the top-k passages' predicted answers is required. A number of different strategies exist for this purpose. A simple but effective method consists of concatenating un-normalised logits of answer distributions over several passages and taking the argmax (Clark and Gardner, 2018; Chen et al., 2017). Reranking methods include learning a passage reranker over the top-k passages and returning the answer from only the highest-reranked passage (Karpukhin et al., 2020), or even learning a dedicated answer reranker over the (c,q,a) triples (Wang et al., 2018d; Iyer et al., 2021). Another empirically effective approach is to learn a reader that aggregates information across paragraphs as part of its architecture (Izacard and Grave, 2021b)

2.4.2.2 Closed-book QA

An alternative modelling approach for ODQA has emerged recently, which solely relies on parametric memory. We refer to this as *closed-book QA*, following the terminology of Roberts et al. (2020). Closed-book QA refers to a class of models where the knowledge source $\mathscr{C}$ is not explicitly available at inference time, in contrast to *open-book* models, such as the retrieve-and-read class.

In section 2.3.2.4 we outlined how pretrained models store relational knowledge from their pretraining corpora in their parameters, and how they can be used to answer cloze questions. Closed-book QA goes one step further, and fine-tunes a seq2seq transformer to directly answer natural questions.

As usual, we assume we have access to a large textual corpus or knowledge source $\mathscr{C}$, and a dataset of question-answer pairs $\mathcal{D} = \{(q_i, a_i)\}_{i=1}^{m}$, answerable from the knowledge in $\mathscr{C}$. We shall first pretrain a large seq2seq model using a denoising objective $p(w_1, \ldots, w_n, |\mathsf{MASK}(w_1, \ldots, w_n); \theta)$ using token sequences $w_1, \ldots, w_n$ drawn from $\mathscr{C}$. Following the argument in section 2.3.2.4, this step encodes relational knowledge present in $\mathscr{C}$ into the parameters θ, and the model should now be able to answer cloze-style questions about knowledge in $\mathscr{C}$.

The second step consists of fine-tuning the model by minimising the negative log likelihood of generating answers given questions:

$$\mathcal{L}_\theta = -\sum_{i,j} \log p(a_{i,j} | a_{i,1}, \ldots, a_{j-1}, q_i; \theta)$$

where $a_{i,j}$ is the j^{th} token of the answer to the i^{th} question in the dataset. This step is intended to convert the model from answering cloze questions to mapping natural questions to their answers. This must be done carefully, as we are risk overwriting knowledge stored in θ.

Closed-book QA models will be smaller, simpler and cheaper at inference time than an architecturally-matched retrieve-and-read model. However, in order to perform well, very high parameter-count closed-book QA models are required in practice (Roberts et al., 2020; Brown et al., 2020; Kirstain et al., 2021).

2.4.2.3 Phrase Index Models

For completeness, we shall briefly sketch a third ODQA model family. Phrase-index models, introduced by Seo et al. (2018) and improved upon by Seo et al. (2019) and Lee et al. (2021), approach span-based ODQA solely as a retrieval problem. We do not directly study these models, but they have a number of intriguing properties, especially when considering latency-accuracy trade-offs. They make for an interesting comparison to QA-pair retrievers, which we introduce in Part III.

Concretely, we assume a large textual knowledge source $\mathscr{C}$ comprised of a very long sequence of N tokens $(w_1, \dots, w_N)$. Given an ODQA task, and dataset $\mathcal{D} = \{(q_i, a_i)\}_{i=1}^{m}$, where answers a are comprised of tokens $(t_1, \dots, t_j)$ up to a maximum length l_t, we assume that every answer exists as a sub-sequence in $\mathscr{C}$ given by indices (a_{st}, a_{en}), i.e.

$$\exists\, (a_{st}, a_{en}) \quad \text{s.t.} \quad (w_{a_{st}}, \dots, w_{a_{en}}) = (t_1, \dots, t_j)\,, \;\; 1 \leq a_{st} \leq a_{en} \leq a_{st} + l_t \leq N$$

In phrase-index models, we will build an index of representations for all phrases in $\mathscr{C}$ up to length l_t (requiring a total of Nl_t vectors), within which every possible answer will be indexed, by definition. Models are learnt using similar techniques to dense retrieval, learning representations for spans of tokens rather than passages. Having built such an index, we can simply encode a question into a query vector, and retrieve an answer phrase using MIPS. This method's key strength is low latency. A downside is the very large number of phrases that must be indexed, which can be somewhat mitigated with compression, decomposition and filtering tricks. Another weakness is that due to the decomposability gap, this approach may have lower accuracy relative to some approaches. In Part III, we introduce a method that shares the low latency of phrase indexers, whilst being more accurate.

2.5 Datasets

Datasets play a vitally important role in modern QA. They influence training and evaluation choices, and shape the evolution of systems and the field as a whole. We shall not attempt a survey of the great variety of QA datasets and the phenomena they test. The interested reader is instead directed to Rogers et al. (2021) and Cambazoglu et al. (2021). We outline only the datasets that form the experimental core of this thesis, first discussing the knowledge source, and then the QA datasets.

2.5.1 Wikipedia as a General-Purpose Knowledge Source

The Wikipedia domain is very popular in modern (English) QA research, and all experimental work in this thesis uses Wikipedia as the underlying knowledge source, either at paragraph level (Part I), or using an entire dump (Parts II-III). Wikipedia is a convenient choice for a knowledge source for several reasons. Firstly, it covers a multitude of different topics, and is more trustworthy, information-dense and cleaner than alternatives like web text, making it well-suited to factual QA. Second, it is large enough to present a difficult scaling problem, without being so big that research is slowed by unmanageable hardware requirements. Third, Wikipedia is less redundant than the web (Mitchell et al., 2009; Chen and Yih, 2020), encouraging more precise models, relying less on information redundancy to derive answers. Forth, Wikipedia is well-annotated with rich metadata, and is multilingual, which can be helpful for both modelling and evaluation. Lastly, Wikipedia is available under CC-BY-SA 3.0 (WikiMedia Foundation, 2021), a license that facilitates data sharing and derivative works, ideal for reproducible, comparable research.

Despite its convenience, there are drawbacks. Wikipedia suffers from under-representation and gender, social, and racial bias issues (Wagner et al., 2016; Adams et al., 2019; Schmahl et al., 2020; Field et al., 2021). As a community, we must remain cognisant of sociological issues, and acknowledge our reliance on Wikipedia will not produce models that perform well on under-represented question distributions. Relatedly, Wikipedia only covers a small fraction of human knowledge, and if we want to build *truly open-domain* models, we will need to move to whole-web and/or multi-source knowledge sources (Oguz et al., 2020; Piktus et al., 2022). Lastly, Wikipedia's homogeneous style and lack of noise means that models developed on it may not be robust, and may transfer poorly to other domains (Wiese et al., 2017; Chung et al., 2018; Talmor and Berant, 2019; Reddy et al., 2020, 2021a).

Evaluation in ODQA research was hampered for a time due to different researchers using different dumps and pre-processing techniques for knowledge sources. The

Statistic	Value
File Size (MiB)	13064
Number of words	2,101,532,400
Number of passages	21,015,324
Number of articles	32,32,908

Table 2.1: Statistics on the Wikipedia open-domain knowledge source used in this thesis

dump we use in thesis was first introduced in Karpukhin et al. (2020) and has since (mostly) been adopted as standard for the ODQA datasets we study. This dump dates from December 2018, and has been extracted from Wikipedia's XML format into plain text using the pipeline of Chen et al. (2017). This dataset *excludes* any semi-structured data, such as tables, lists and info-boxes, as well as disambiguation pages, leaving only the textual content. This corpus is then split into 21 million disjoint text passages, each 100 words in length. Finally, the passage is prepended with the title of the article it comes from. Statistics can be found in Table 2.1.

2.5.2 Question Answering Datasets

2.5.2.1 Questioner Intent

Before describing the QA datasets themselves, it is instructive to define two broad classes of dataset, based on questioner intent. We borrow the terminology of Rodriguez and Boyd-Graber (2021), who, in turn, borrow from IR concepts (Voorhees, 2002b; Robertson, 2008). Rodriguez and Boyd-Graber argue that there are two high-level QA intents, which affect how questions should be processed, and how answers should be evaluated. The first, referred to as the *Cranfield* paradigm, refers to building systems tasked with serving human information needs. Here, questions will be information-seeking, from a questioner with a genuine information need that they hope the answerer will satisfy. For the second, the *Manchester* paradigm, the intent is *not* to fulfil an information need, but rather to challenge, test or probe the answerer on their reasoning abilities and their command of their knowledge.

A key difference between Cranfield and Manchester paradigms is whether the questioner knows the answer to their question before they ask it. This has deep implications for how questions are phrased, and how they should be processed.

Question: How many nations control this region in total?
Answer: Nine
Context: The Amazon rainforest, also known in English as Amazonia or the Amazon Jungle, is a moist broadleaf forest covering most of the Amazon basin of South America. This basin encompasses 7 million KM^2 [...]. The region includes territory belonging to nine nations. The majority of the forest is contained within Brazil, with 60% of the rainforest, followed by Peru with 13%, Colombia with 10%, and with minor amounts in [...]. States or departments in four nations contain "Amazonas" in their names [...]

Figure 2.1: Example from SQuAD, with answer highlighted in grey

Typically Cranfield tasks and datasets will have questions harvested from query logs from existing, practical QA systems. Since the questioner does not know the answer to their question, they will often ask *under-specified* questions, where the answerer must make inferences about what answer best serves the questioner's need. Cranfield questioners may formulate questions to increase the odds of the answerer getting the question correct, perhaps by including helpful keywords.

Since the primary goal of Manchester-style QA is to probe and challenge the answerer, these tasks are often typified by questions sourced from trivia competitions, e.g. *Jeopardy!* (Ferrucci et al., 2010) and Quizbowl (Rodriguez et al., 2019), or *adversarial* questions, specifically formulated to exploit weaknesses in existing models (Jia and Liang, 2017; Wallace et al., 2019; Bartolo et al., 2020) .

Improvements to models on paradigm may improve their performance in the other, and both paradigms require solving similar core problems. However, an over-focus on one paradigm may prove detrimental in the other – e.g being more robust to the extreme tail of unnatural adversarial questions may lead to better Manchester-oriented results, but may sacrifice accuracy on more probable questions.

2.5.2.2 SQuAD v1

In Part I, we concentrate on extending RC capabilities on the span-extraction dataset SQuAD. SQuAD is a prototypical modern RC dataset and is comprised of over 100K context, question, answer triples. Contexts are paragraphs from featured Wikipedia articles, and questions and answers were obtained by presenting paid crowd-workers with a paragraph, and instructing them to formulate a natural language question, and to highlight an answer span within the text. SQuAD is thus

Dataset	Question source	Answer Source	Answer Format	Sufficient Knowledge Source	Knowledge Source used in Practice	Format
SQuAD-v1	Crowdworkers	Crowdworkers	Span	Wikip. Passage	Wikip. Passage	Span-RC
MLQA	Crowdworkers	Crowdworkers	Span	Wikip. Passage	Wikip. Passage	Span-RC
TriviaQA	Trivia Websites	Trivia Websites	Text / Span†	Wikipedia	Wikipedia (Text only)	ODQA
NQ-open	Search Logs	Tool-assisted Crowdworkers	Text / Span	Wikipedia	Wikipedia (Text only)	ODQA
WebQuestions	Search Logs	Tool-assisted Crowdworkers	Text / Span†	Freebase	Wikipedia (Text only)	ODQA
CuratedTREC	Search Logs	Tool-assisted Crowdworkers	Text / Span†	Wikipedia	Wikipedia (Text only)	ODQA
MsMarco NLG	Search Logs	Crowdworkers	Free-form Text	Web Passage	Not Attempted	Abstractive-RC
MsMarco NLG-open	Search Logs	Crowdworkers	Free-form Text	Web subset	Wikipedia (Text only)	Abstractive-ODQA

Dataset	Passage-grounded Answer Annotations	Ave. Question Length (words)	Ave. Answer Length (words)	Ave. Knowledge Source Length (words)	# Train Instances	# Dev Instances	# Test Instances	Eval Metric
SQuAD-v1	Hand-annotated	10.1 ± 3.6	3.1 ± 3.3	120.2 ± 50.2	87599	10570	9616	EM,F1
MLQA	Hand-annotated	9.4 ± 3.3	3.2 ± 3.6	169.4 ± 144.2	-	504→1148	4517→11590	EM,F1
TriviaQA	Distant Supervision	14.1 ± 7.2	2.5 ± 1.5	2,101,532,400	79,168	8757	3610	EM
NQ-open	Hand-annotated	9.1 ± 1.6	2.1 ± 1.0	2,101,532,400	78,785	8837	11313	EM
WebQuestions	Distant Supervision	6.8 ± 1.6	2.4 ± 2.4	2,101,532,400	3,417	361	2032	EM
CuratedTREC	Distant Supervision	7.5 ± 3.0	-	2,101,532,400	1,353	133	694	REGEX
MsMarco NLG	Hand-annotated	6.0 ± 2.4	14.6 ± 9.2	524 ± 129	153726	12468	101093	Bleu,Rouge
MsMarco NLG-open	None	6.0 ± 2.4	14.6 ± 9.2	2,101,532,400	153726	12468	101093	Bleu,Rouge

Table 2.2: Summaries and statistics for select datasets featuring in this thesis. † indicates answers are not technically spans, but can be modelled as spans without significant losses. * indicates statistic calculated on English portion.

Dataset	Question	Answer
CuratedTREC	Where does the vice president live when in office?	U.S\s?. Naval Observatory
WebQuestions	what is the state flower of arizona?	Saguaro
TriviaQA	who was britain's only track and field gold medallist at the 1972 olympics	Mary Peters
NaturalQuestions	whats the difference between tomato paste and tomato puree	consistency

Table 2.3: Example question-answer pairs from popular open-domain QA datasets

"Manchester-style" and is evaluated using EM and mean F1 (see section 2.6).

A typical instance from SQuAD is shown in Figure 2.1, with statistics in Table 2.2. A later extension to the dataset, SQuAD v2 (Rajpurkar et al., 2018) includes unanswerable questions, which we do not use in this thesis.

Limitations Whilst SQuAD is undoubtedly useful, there are a number of limitations. There is a high lexical overlap between the question and the sentence that surrounds the answer (Weissenborn et al., 2017b; Sugawara et al., 2018). This is due to the annotation procedure, which incentives writing questions quickly, and lack of intrinsic motivation for annotators to pose, lexically-diverse questions. As a result, models may learn to overly-rely on lexical matching to locate answers, which can make them brittle (Jia and Liang, 2017). Moreover, models also learn to pick up on type consistency clues – a question containing the word "where" can often be answered by finding a location mention in the passage, without requiring a deep understanding of question meaning (Sugawara et al., 2018). Such basic answering strategies are never-the-less "a good start", especially if they can be induced without thousands of hours (and dollars) of training set annotation. We present an approach to learn these reading skills in an unsupervised way in chapter 3.

An open-domain version of SQuAD (SQuAD-open) has seen some use, whereby systems are required to answer SQuAD questions using the whole of Wikipedia, rather than a single passage (Chen et al., 2017; Lee et al., 2019a; Karpukhin et al., 2020, inter alia.). However, this use-case has some severe issues, and there are many superior alternatives available. First, SQuAD questions often do not make sense without their context passage. For example, the SQuAD question "what did this concept contradict?" cannot have a good answer in the open-domain case. Clark and Gardner (2018) estimate 33% of SQuAD questions are similarly context-dependent. Due to high lexical overlap between questions and contexts, and due to SQuAD having many questions annotated on a small number of passages, there is a high bias in SQuAD-open. Accordingly, simple term-based retrievers perform spuriously well (Lee et al., 2019a).

2.5.2.3 CuratedTREC (CT) and WebQuestions (WQ)

CT (Baudis and Sedivy, 2015) and WQ (Berant et al., 2013) are common ODQA datasets. They are smaller, with comparatively older questions than alternatives. See Tables 2.2 and 2.3 for statistics and examples. Both contain questions originally from search engine query logs, and are Cranfield-style tasks.

CT is a compilation of questions from TREC QA tracks between 1999 and 2003. CT's answers are annotated by crowdworkers, and questions are short, relatively entity-centric, and intended to be answerable from Wikipedia. WQ's questions were also answered by crowdworkers but answers are restricted to be lexicalized entities in a structured KB, Freebase (Bollacker et al., 2008). Whilst CT was specifically created for textual ODQA, WQ was originally intended for semantic parsing, but has been re-purposed. Both datasets lack passage-grounded answer annotations i.e. we do not know precisely which passages contain sufficient information to answer the question. Instead, it is common to rely on distant supervision, whereby noisy passage-grounded answer annotations are obtained by retrieving passages using e.g. BM25, and taking the highest-ranked passages which contain the answer.

2.5.2.4 TriviaQA (TQA)

TQA (Joshi et al., 2017) is a popular ODQA dataset, comprised of questions and answers scraped from trivia websites.[10] Statistics can found in Table 2.2, and examples in Table 2.3. Since the questions are from trivia competitions, they are designed to be challenging (but not too challenging) for human quizzers. As the questioner

[10] TriviaQA also has an RC task, which we do not consider in this thesis

knows the answer to the question they pose, these questions are not representative of what information-seeking users may ask, and TriviaQA is thus Manchester-style. All answers in TriviaQA are restricted to be Wikipedia entities, guaranteeing that they exist in Wikipedia. Like CT and WQ, passage-grounded answer annotations are not available, and distant supervision is often employed. TriviaQA is also much larger than CT and WQ, (100K QA-pairs verses 5K and 2K respectively), which allows for more supervision and lower-variance evaluations.

2.5.2.5 NaturalQuestions (NQ)

NQ (Kwiatkowski et al., 2019) is a recent RC dataset. NQ was carefully constructed to avoid the issues with the first wave of large RC datasets like SQuAD. Questions are sourced from Google query logs, and, like those from WQ and CT, are genuine, information-seeking questions, making NQ Cranfield-style. Context documents are whole Wikipedia articles, and answers are annotated by crowdworkers at two levels of granularity: paragraph level ‘long answers’ and ‘short answer’ spans. Substantial curation and annotation costs were invested to produce NQ, which, coupled with its careful design and size make it a strong general-domain RC dataset.

Whilst originally created for RC, it has since been re-purposed for ODQA, first by Lee et al. (2019a). It is this ODQA version which we use in thesis. To convert NQ for ODQA, yes/no and unanswerable questions were removed, as were answers longer than 5 tokens. Like the other ODQA datasets, questions tend to be factoid, and answers are usually noun-phrases, dates, or entities. Passage-grounded answer annotations *are* available in NQ, which is useful both for additional supervision and evaluation options relative to the other datasets.

2.6 Evaluation

The evaluation problem amounts to deciding whether an answer $\hat{a}$ produced by a system is a correct answer to a question q. This is a challenging problem in-and-of itself. The gold standard is to use human evaluators, a strategy employed in TREC competitions, and more recently, the EfficientQA NeurIPS competition (Min et al., 2021). Such evaluations are of immense value and should be carried out frequently. However, this is expensive, time-consuming, and presents reproducibility challenges. In practice, automatic evaluation procedures are also required.

To avoid the problem of automatically verifying the correctness of an answer, we instead tackle the easier task of assessing whether a predicted answer $\hat{a}$ is equivalent to a gold reference answer a. Note this will *not* capture cases where $\hat{a}$ is a correct but different answer to a, which we discuss further below.

2.6.1 Automatic Metrics

The choice of evaluation metric is defined by the answer format of the task. For multiple-choice QA, evaluation is straightforward – simple classification accuracy or F1 is sufficient. For span-based RC and ODQA, it has become common to use the evaluation procedure of Rajpurkar et al. (2016). Here, answer strings are lightly normalised by lowercasing, removing articles and punctuation and then whitespace-tokenised. Then, for a predicted and reference answer $\hat{a}$ and a, comprised of tokens $(\hat{a}_1, \ldots, \hat{a}_m)$ and $(a_1, \ldots, a_m)$, we define the *exact match* (EM) score as

$$\mathsf{ExactMatch}(\hat{a}, a) = \mathbb{1}_{\hat{a}=a}$$

in other words, does the (normalised) predicted answer exactly match the reference answer? This is a harsh metric, since all credit is lost if the answer includes or misses out a single token not present in the reference. For example, the predicted answer "King Henry VIII" would receive no credit for the reference answer "Henry VIII". Thus, a softer metric is also often used, the bag-of-tokens F1:

$$\mathsf{Prec}(\hat{a}, a) = \frac{|\{\hat{a}_1, \ldots \hat{a}_m\} \cap \{a_1, \ldots a_m\}|}{|\{\hat{a}_1, \ldots \hat{a}_m\}|} \quad \mathsf{Rec}(\hat{a}, a) = \frac{|\{\hat{a}_1, \ldots \hat{a}_m\} \cap \{a_1, \ldots a_m\}|}{|\{a_1, \ldots a_m\}|}$$

$$\mathsf{F1}(\hat{a}, a) = \frac{2 \cdot \mathsf{Prec}(\hat{a}, a) \cdot \mathsf{Rec}(\hat{a}, a)}{\mathsf{Prec}(\hat{a}, a) + \mathsf{Rec}(\hat{a}, a)}$$

It is common to provide several annotated gold answer references for test sets, whereby final EM and F1 score for a prediction is given by its maximum score across the answer references. These instance-level scores are aggregated into an overall test-set score by simple macro-average. For CT, test-set answer annotations come in the form of regular expressions, and predictions receive credit only if they match the regular expressions. Lastly, for longer-answer and free-form QA, it is common to use BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004).

2.6.2 Limitations of Standard Evaluation Practice

The automatic evaluation protocol is appealing due to its simplicity and efficiency. It does, however, have a number of issues. Firstly, most datasets do not contain more than a handful of answer references for each question, which can lead to a high number of false negatives, especially in ODQA (Roberts et al., 2020; Si et al., 2021). Ensuring that enough references are provided to cover the lexical variability of the correct answer, whilst still being precise is a difficult annotation challenge. A separate but related issue is that of test questions which have alternative, un-annotated correct answers. This usually occurs when questions are under-specified. Min et al. (2020) use the term *ambiguous questions*, providing the NQ example, `when did harry potter and the sorcerers stone come out?`, which, in reality has two valid answers – the premiere was 4th November 2001, and the general cinema release was on 16th. However, only the premiere is annotated as an answer, despite both being valid answers. This is relatively common when questioners do not know the answer to the question they are asking.

Nevertheless, assuming the answer annotations that are available *are* of high quality, our evaluation metrics are precise, i.e. if automatic evaluation indicates a question is answered correctly, it is very likely to be true. Throughout this thesis, we will state that models achieve a certain “accuracy” on a test set, based on automatic evaluation scores. We should bear in mind that terminology like “accuracy” in this context is a short-hand for saying that such scores constitute a *lower bound* of their true accuracy. Put simply, an EM score of 45% does not imply a model will answer ∼45% of questions correctly. A recent human evaluation found models to have absolute accuracies 10%-21% higher than EM scores (Min et al., 2021).

Lastly we note that the community has a tendency to focus on overall, i.i.d. test-set scores, which can be misleading. We shall explore such a case in chapter 7. Adversarial and "challenge" test sets, which share the same evaluation methodology, but expose weaknesses are an increasingly popular way of thoroughly evaluating model behaviour (Lehmann et al., 1996; Jia and Liang, 2017; Belinkov and Glass, 2019; Bartolo et al., 2020; Ribeiro et al., 2020; Sciavolino et al., 2021, inter alia.).

2.7 Miscellany

2.7.1 A Note on Answerability

In the above, we maintained that the QA tasks we consider are only defined over questions that are answerable given the knowledge source. It is worth noting however that some RC tasks, not covered in this thesis, *do* include unanswerable questions as part of the task. Such efforts have not yet hit mainstream ODQA, where answerability is guaranteed by construction (NQ) or by filtering potential unanswerables (TQA), or simply by assumption (CuratedTREC, WQ). In practice, unanswerable questions may exist due to pragmatic implementation choices. For example, we use only the text of Wikipedia, which may miss answers from tables. Coupled with the existence of ambiguous/under-specified questions, we note that, in practice, full answerability is an assumption only. Table 2.2 has columns describing what knowledge source is sufficient to answer every question in a dataset vs. what we use in practice in our experiments. Finally, even if an answer exists, it doesn't guarantee that an model will be able to find it. In cases where a model has failed to find good evidence for an answer, it may be better to refuse to answer, rather than return likely-incorrect guesses. We study such behaviour in chapter 8.

2.7.2 Dialogue

Some consider QA to be a special case of dialogue, consisting of a single utterance exchange. We do not tackle dialogue in this thesis, but agree Cranfield-style QA should be a collaboration between user and model. A potential solution to ambiguous questions is to generate clarification questions in order to find the best answer (Saeidi et al., 2018; Min et al., 2020). A number of *conversational QA* tasks have been proposed for RC (Saeidi et al., 2018; Choi et al., 2018; Reddy et al., 2019), and more recently for ODQA (Qu et al., 2020; Anantha et al., 2021)

2.7.3 Relationship between QA & Knowledge-Intensive NLP

In chapter 6 we tackle tasks that we refer to as *knowledge-intensive NLP* tasks. This term describes NLP tasks with emphasis on being able to leverage large amounts of knowledge. It is often defined as tasks where the average human would require access to the knowledge source at inference time. ODQA is a typical Knowledge Intensive NLP task. Other tasks identified by Petroni et al. (2021) employing a Wikipedia knowledge source are fact checking (Thorne et al., 2018), slot-filling (Levy et al., 2017; Elsahar et al., 2018), entity linking (Hoffart et al., 2011; Guo and Barbosa, 2018) and some forms of dialogue (Dinan et al., 2019). These tasks share common prerequisite skills, so models that tend to do well on ODQA will likely also perform well on other knowledge-intensive tasks. Indeed, many of these tasks could be reduced to ODQA with suitable transformations of their inputs and there is a deep connection between ODQA and knowledge-intensive NLP in general.

2.7.4 Desirable Properties of Question Answering systems

When comparing different QA systems, it is tempting to simply compare their accuracies and conclude that one model is "better" than another. However, such an evaluation is misleading, and ignores a wide range of important criteria. For example, for user-facing systems, the better model may be the one with lower latency, or cheaper inference requirements, rather than higher accuracy. If performing purer Manchester-style research, models that require less training data may be better than models that eventually outperform them given more data. Or, is a monolingual model that scores 3% higher on a test set better than one that can answer questions in many languages? Comparing models is nuanced and must be done carefully.

For many criteria such as latency and size, models may come with hyperparameters that allow us to tune the tradeoffs, turning point-wise comparisons into operating curves. Accuracy is important, but we should not necessarily prioritise it at the expense of other factors. For example, it is valuable for an ODQA system to be explainable, or to provide evidence or *providence* for its answers. Retrieve-and-read models have a natural mechanism for this, since we can examine the retrieved passages that the answers are conditioned on. Closed-book QA models are black-boxes, with no straightforward providence mechanism. Summing up, a good QA system will be accurate, but a great one will be accurate, multilingual, low-latency, small, require little training data, and provide evidence for its answers.

Part I

Reading Comprehension without Task Annotations

"Self-education is, I firmly believe, the only kind of education there is."

Isaac Asimov

Chapter 3

Unsupervised Reading Comprehension by Cloze Translation

In Part I, we shall concern ourselves with the span-extractive RC task, introduced in section 2.4.1. For common RC datasets like SQuAD (section 2.5.2.2), fine-tuned pretrained models achieve results on par with humans on in-domain test sets. However, this is predicated on the availability of large amounts of hand-annotated training data. Unfortunately, for new domains or languages, collecting such training data is non-trivial, time-consuming and expensive. Reducing, or even entirely removing the requirement for hand-annotated RC training data would greatly increase applicability and access to RC technologies, which are not only useful in-and-of-themselves, but also form important components of other models. This is the motivation underpinning Part I.

In this chapter, we make a start on this agenda by considering the question: “What if no training data was available *at all*?” Concretely, we shall investigate the feasibility of *unsupervised* RC, a novel setting in which no aligned question, context and answer data is available. This extreme setting, whilst somewhat unrealistic in a practical sense, is a useful exercise for empirically assessing to what extent hand-annotated RC training data is really needed. Moreover, by preventing ourselves from using aligned data, we may gain a new perspective on the RC task. For example, by assessing what basic behaviours we can – and cannot – induce in an RC model without needing training data, we may learn more about where the real challenges and subtleties of RC lie. Lastly, if we *can* achieve some RC competency, then such a model may be better able to use any small amounts of annotated data that do become available, such as in few-shot or semi-supervised settings.

Bigger Picture: This chapter focuses entirely on the task of RC. Ultimately, while RC is a valuable in-and-of-itself, we are more interested in more general ODQA settings, where context documents are not provided. That said, doing well on RC is a prerequisite to strong ODQA, and, by initially studying the simpler task, we can make faster progress without needing to handle the additional complexity of ODQA. Moreover, RC components play an important role in ODQA models (especially retrieve-and-read models, see sec. 2.4.2.1), and techniques developed in RC could be readily applied to ODQA settings. This chapter is devoted to understanding what QA behaviours we can achieve without annotate data, which the two chapters proceeding it will build upon. Finally, ideas relating to automatic question generation for QA, which we begin to use in this chapter, will return in chapter 8, where we will use them in an open-domain generation setting to induce much faster, better calibrated and more flexible ODQA models. Additional commentary on the connections between this chapter and the wider body of work in the thesis can be found in the conclusion of this chapter (sec. 3.4) and the thesis conclusion, chapter 9.

The material in this chapter first appeared in:

> **Patrick Lewis**, Ludovic Denoyer, and Sebastian Riedel. 2019. Unsupervised Question Answering by Cloze Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*

> *Individual Contributions: The idea of applying UMT for question generation was proposed by co-authors. The final conceptual design was a close collaboration, with substantial con-*

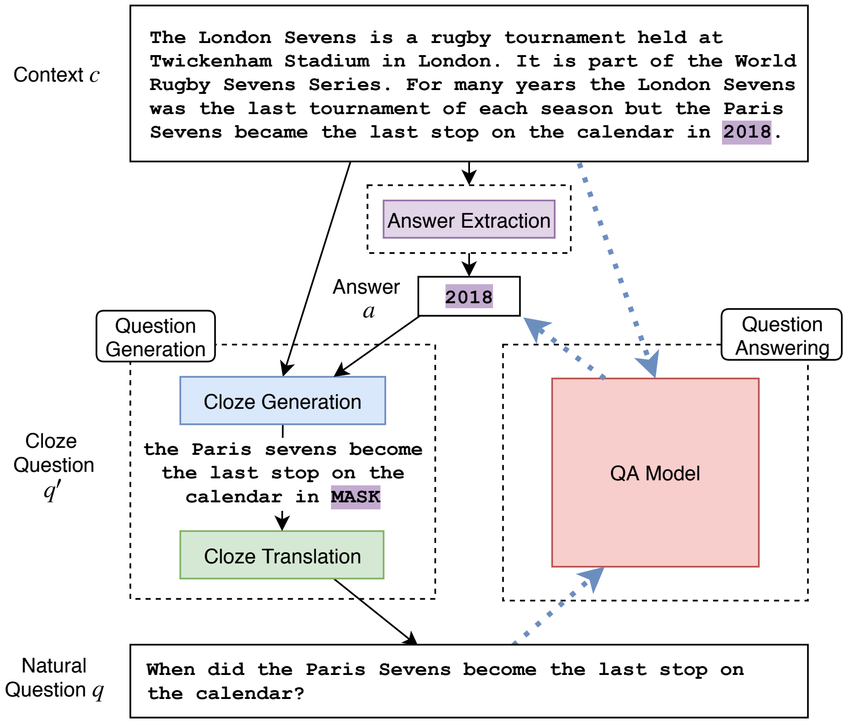

Context c
The London Sevens is a rugby tournament held at Twickenham Stadium in London. It is part of the World Rugby Sevens Series. For many years the London Sevens was the last tournament of each season but the Paris Sevens became the last stop on the calendar in 2018.
Answer Extraction
Answer a
2018
Question Generation
Cloze Generation
Cloze Question q'
the Paris sevens become the last stop on the calendar in MASK
Cloze Translation
Question Answering
QA Model
Natural Question q
When did the Paris Sevens become the last stop on the calendar?

Figure 3.1: A schematic of our approach. The right side (dotted arrows) represents traditional RC. We introduce unsupervised data generation (left side, solid arrows), which we use to train standard RC models

tributions from the thesis author. All experiments and analysis were devised and performed by the thesis author, as was the majority of the writing of the original paper.

3.1 Overview

We propose to tackle the unsupervised RC task by reducing it to unsupervised question generation – if we had a method, without using RC supervision, to generate accurate questions given a context document, we could then use the generated questions to train an RC model. This approach allows us to directly leverage recent progress in RC, such as model architectures and pretrained RC models. This framework is attractive in both its flexibility and extensibility. In addition, our method can be used to augment training data in few-shot and semi-supervised settings.

Our proposed method, shown schematically in Figure 3.1, generates RC training data in three steps:

1. We sample a paragraph in a target domain – in our case, English Wikipedia.

2. We sample from a set of candidate answers within that context, using pre-trained components (NER or noun chunkers) to identify such candidates. These require supervision, but no aligned q,a or q,c data. Given a candidate answer and context, we can extract "fill-the-blank" cloze questions.

3. Finally, we convert cloze questions into natural questions using an unsupervised cloze-to-natural question translator.

The conversion of cloze questions into natural questions is the most challenging of these steps. While there exist sophisticated rule-based systems (Heilman and Smith, 2010) to transform statements into questions (for English), we find their performance to be empirically weak as a source of supervision data for RC (see Section 3.3). Moreover, for specific domains or other languages, a substantial engineering effort will be required to develop similar algorithms. Also, whilst supervised models exist for this task, they require the type of annotation unavailable in this setting (Du et al. 2017; Du and Cardie 2018; Hosking and Riedel 2019, *inter alia*). We overcome this issue by using *unsupervised machine translation* which was described in detail in section 2.3.1.3 in chapter 2. In particular, we collect a large corpus of natural questions and an *unaligned* corpus of cloze questions, and train a seq2seq model to map between natural and cloze question domains using a combination of back-translation and de-noising.

In our experiments, we find that in conjunction with modern RC model architectures, unsupervised RC can lead to performances surpassing early supervised approaches (Rajpurkar et al., 2016). We show that forms of cloze "translation" that produce (unnatural) questions via word removal and flips of the cloze question lead to better performance than an informed rule-based translator. Moreover, the unsupervised seq2seq model outperforms both the noise and rule-based system. Lastly, we show that our method can be used in a few-shot learning setting, obtaining 59.3 F1 with 32 labelled examples, compared to 40.0 F1 without our method.

To summarise, this chapter makes the following contributions: i) The first approach for unsupervised RC, reducing the problem to unsupervised cloze translation, using methods from unsupervised machine translation ii) Extensive experiments testing the impact of various cloze question translation algorithms and assumptions iii) Experiments demonstrating the application of our method for few-shot RC.

3.2 Unsupervised RC

We propose to address unsupervised RC in a two stage approach. We first develop a generative model $p(q,a,c)$ using no (RC) supervision, and then train a discriminative reader $p_r(a|q,c)$ using p as training data generator. The generator $p(q,a,c) = p(c)p(a|c)p(q|a,c)$ will generate data in a “reverse direction”, first sampling a context via $p(c)$, then an answer within the context via $p(a|c)$ and finally a question for the answer and context via $p(q|a,c)$. In the following we present variants of these components.

3.2.1 Context and Answer Generation

Given a corpus of documents our context generator $p(c)$ uniformly samples a paragraph c of appropriate length from any document, and the answer generation step creates answer spans a for c via $p(a|c)$. This step incorporates prior beliefs about what constitutes good answers. We propose two simple variants for $p(a|c)$:

Noun Phrases We extract all noun phrases from paragraph c and sample uniformly from this set to generate a possible answer span. This requires a chunking algorithm for our language and domain.

Named Entities We can further restrict the possible answer candidates and focus entirely on named entities. Here we extract all named entity mentions using an NER system and then sample uniformly from these. Whilst this reduces the variety of questions that can be answered, it proves to be empirically effective as discussed in Section 3.3.2.

3.2.2 Question Generation

Arguably, the core challenge in RC is modelling the relation between question and answer. This is captured in the question generator $p(q|a,c)$ that produces questions from a given answer in context. We divide this step into two steps: cloze generation $q' = \mathsf{cloze}(a,c)$ and translation, $p(q|q')$.

3.2.2.1 Cloze Question Generation, $q' = \text{cloze}(a,c)$

Here, we follow a similar process to that described in section 2.3.1.1. In the first step, we reduce the scope of the context to roughly match the level of detail of actual questions in RC tasks. A natural option is to use the sentence around the answer. Using the context and answer from Figure 3.1, this might leave us with the sentence "For many years the London Sevens was the last tournament of each season but the Paris Sevens became the last stop on the calendar in ____". We can further reduce length by restricting to sub-clauses around the answer, based on access to an English syntactic parser, leaving us with "the Paris Sevens became the last stop on the calendar in ____".

3.2.2.2 Cloze Translation, $p(q|q')$

Once we have generated a cloze question q' we translate it into a form closer to what we expect in real RC tasks. We explore four approaches here:

Identity Mapping We consider that cloze questions themselves provide a signal to learn some form of RC behaviour. To test this hypothesis, we use the identity mapping as a baseline for cloze translation. To produce "questions" that use the same vocabulary as real RC tasks, we replace the mask token with a wh*-word, either randomly chosen, or with a simple heuristic (see Section 3.2.4).

Noisy Clozes One way to characterise the difference between cloze and natural questions is as a form of perturbation. To improve robustness to perturbations, we can inject noise into cloze questions. We implement this as follows. First, we delete the mask token from cloze q', apply a noise function, and prepend a wh*-word (randomly or with the heuristic in Section 3.2.4) and append a question mark. The noise function consists of word dropout, word order permutation and word masking. The motivation is that, at least for SQuAD, it may be sufficient to simply learn to identify a span surrounded by high n-gram overlap to the question, with a tolerance to word order perturbations.

Rule-Based Turning an answer embedded in a sentence into a (q,a) pair can be understood as a syntactic transformation with wh*-word-movement and a type-dependent choice of wh*-word. For English, off-the-shelf rule-based software exists for this purpose. We use the popular statement-to-question generator from Heilman and Smith (2010) which was described in detail in section 2.3.1.1

Unsupervised Machine Translation (UMT) The above approaches either require substantial engineering and prior knowledge or are still far from generating natural-looking questions. We propose to overcome both issues through unsupervised training of a seq2seq model that translates between cloze and natural questions, which we describe in detail in Section 3.2.4.

3.2.3 Question Answering

RC amounts to finding the best answer a given question q and context c. We have at least two ways to achieve this using our generative model:

Training a separate RC system The generator is a source of training data for any RC architecture at our disposal. Whilst the data we generate is unlikely to match the quality of real RC data, we hope QA models will learn basic QA behaviours.

Using Posterior Another way to extract the answer is to find a with the highest posterior $p(a|c,q)$. Assuming uniform answer probabilities conditioned on context $p(a|c)$, this amounts to calculating $\arg\max_{a'} p(q|a',c)$, i.e. testing how likely each possible candidate answer span is to have generated the question, a similar method to the supervised approach of Lewis and Fan (2018).

3.2.4 Unsupervised Cloze Translation

To train a seq2seq model for cloze translation we borrow ideas from unsupervised Machine Translation (UMT) which was described in detail in section 2.3.1.3. In order to apply this technique, no parallel data is required, only *non-parallel* corpora of source and target "language" sentences. In our setting, we aim to learn a function which maps between the question (target) and cloze question (source) domains. For this, we need corpora of cloze questions $\mathcal{D}_s$ and natural questions $\mathcal{D}_t$.

High Level Answer Category	Named Entity Labels	Most appropriate wh*
PERSON/NORP/ORG	PERSON, NORP, ORG	Who
PLACE	GPE, LOC, FAC	Where
THING	PRODUCT, EVENT, WORKOFART, LAW, LANGUAGE	What
TEMPORAL	TIME, DATE	When
NUMERIC	PERCENT, MONEY, QUANTITY, ORDINAL, CARDINAL	How much / How many

Table 3.1: High level answer categories for the different named entity labels

Cloze Corpus We create the cloze corpus $\mathcal{D}_s$ by applying the cloze generation procedure described above in Section 3.2.2.2. Specifically we consider Noun Phrase (NP) and Named Entity mention (NE) answer spans, and cloze question boundaries set either by the sentence or sub-clause that contains the answer.[1] We extract 5M cloze questions from randomly sampled Wikipedia paragraphs, and build a corpus $\mathcal{D}_s$ for each choice of answer span and cloze boundary technique. See Appendix C.1 for exhaustive details.

Question Corpus We mine questions from English pages from a recent Common-Crawl[2] dump using simple selection criteria: We select sentences that start in one of a few common wh*-words, ("how much", "how many", "what", "when", "where" and "who") and end in a question mark. We reject questions that have repeated question marks or "?!", or are longer than 20 tokens. This process yields over 100M English questions when de-duplicated. Corpus $\mathcal{D}_t$ is created by sampling 5M questions such that there are equal numbers of questions with each wh*-word.

Following the unsupervised MT procedure (section 2.3.1.3), we then use $\mathcal{D}_s$ and $\mathcal{D}_t$ to train translation models $p_{s\rightarrow t}(q|q')$ and $p_{t\rightarrow s}(q'|q)$ which translate cloze questions into natural questions and vice-versa. At inference time, natural questions are generated from cloze questions as $\arg\max_q p_{s\rightarrow t}(q|q')$. Additional experimental details can be found in Appendix C.2.

Wh* heuristic In order to provide an appropriate wh*-word for our "identity" and "noisy cloze" baseline question generators, we implement a simple heuristic that maps each answer type to the most appropriate wh*-word, as shown in Table 3.1. For example, the "TEMPORAL" answer type is mapped to "when". During experi-

[1] We use SpaCy for Noun Chunking and NER, and AllenNLP for the parser Stern et al. (2017).

[2] http://commoncrawl.org/

ments, we find that the unsupervised MT translation functions sometimes generate inappropriate wh*-words for the answer entity type, so we also experiment with applying the wh* heuristic to these question generators. For the MT models, we apply the heuristic by prepending target questions with the answer type token mapped to their wh*-words at training time – e.g. questions that start with "when" are prepended with the token "TEMPORAL".

3.3 Experiments

We want to explore what QA performance can be achieved without using *aligned* RC data, and how this compares to supervised learning and other approaches which do not require training data. Furthermore, we seek to understand the impact of different design decisions upon QA performance of our system and to explore whether the approach is amenable to few-shot learning when only a few annotated instances are available. Finally, we also wish to assess whether unsupervised MT can be used as an effective method for question generation.

3.3.1 Unsupervised RC Experiments

For the synthetic dataset training method, we consider two QA models: finetuning BERT (Devlin et al., 2019) and BiDAF + Self Attention (Clark and Gardner, 2018). For the posterior maximisation method, we extract cloze questions from both sentences and sub-clauses, and use the MT models to estimate $p(q|c,a)$. We evaluate using the standard EM and F1 (section 2.6).

As we cannot assume access to a development dataset when training unsupervised models, we halt training when F1 score on a held-out set of synthetic QA data plateaus. We do, however, use the SQuAD development set to assess which model components and factors are important (Section 3.3.2). To preserve the integrity of the SQuAD test set, we only submit our best system to the test server.

We shall compare our results to some published baselines. Rajpurkar et al. (2016) use a supervised logistic regression model with feature engineering, and a sliding window approach that finds answers using word overlap with the question. Kaushik

Unsupervised Models	EM	F1
Ours: BERT-Large Unsupervised RC (ensemble)	**47.3**	**56.4**
Ours: BERT-Large Unsupervised RC (single)	44.2	54.7
BiDAF+SA (Dhingra et al., 2018)	3.2†	6.8†
BiDAF+SA (Dhingra et al., 2018)‡	10.0*	15.0*
BERT-Large (Dhingra et al., 2018)‡	28.4*	35.8*
Baselines	EM	F1
Sliding window (Rajpurkar et al., 2016)	**13.0**	**20.0**
Context-only (Kaushik and Lipton, 2018)	10.9	14.8
Random (Rajpurkar et al., 2016)	1.3	4.3
Fully Supervised Models	EM	F1
BERT-Large (Devlin et al., 2019)	**84.1**	**90.9**
BiDAF+SA (Clark and Gardner, 2018)	72.1	81.1
Log. Reg. + FE (Rajpurkar et al., 2016)	40.4	51.0

Table 3.2: Our best performing unsupervised RC models compared to various baselines and supervised models. * results on SQuAD dev set. † results on non-standard test set created by Dhingra et al. (2018). ‡ our re-implementation

and Lipton (2018) train (supervised) models that disregard the input question and simply extract the most likely answer span from the context. To our knowledge, ours is the first work to deliberately target unsupervised RC on SQuAD. Dhingra et al. (2018) focus on semi-supervised QA, but do report a unsupervised number. For fair comparison, we re-implement their approach using their publicly available data, and train a BERT-Large variant. Their approach also uses cloze questions, but without translation, and heavily relies on the special structure of Wikipedia articles.

Our best approach attains 54.7 F1 on the SQuAD test set; an ensemble of 5 models achieves 56.4 F1. Table 3.2 shows the result in context of published baselines and supervised results. Our approach significantly outperforms baseline systems and Dhingra et al. (2018) and surpasses early supervised methods.

3.3.2 Ablation Studies and Analysis

To understand the different contributions to the performance, we undertake an ablation study. All ablations are evaluated using the SQUAD development set. We report ablation numbers using BERT-Base and BiDAF+SA, and our best performing setup is then used to fine-tune a final BERT-Large model, which is the model in Table 3.2. All experiments with BERT-Base were repeated with 3 seeds, we re-

Cloze Answer	Cloze Boundary	Cloze Translation	Wh* Heuristic	BERT-Base EM	BERT-Base F1	BiDAF+SA EM	BiDAF+SA F1	Post Max. EM	Post Max. F1
NE	Sub-clause	UMT	✓	**38.6**	**47.8**	**32.3**	**41.2**	**17.1**	**21.7**
NE	Sub-clause	UMT	×	36.9	46.3	30.3	38.9	15.3	19.8
NE	Sentence	UMT	×	32.4	41.5	24.7	32.9	14.8	19.0
NP	Sentence	UMT	×	19.8	28.4	18.0	26.0	12.9	19.2
NE	Sub-clause	Noisy Cloze	✓	36.5	46.1	29.3	38.7	-	-
NE	Sub-clause	Noisy Cloze	×	32.9	42.1	26.8	35.4	-	-
NE	Sentence	Noisy Cloze	×	30.3	39.5	24.3	32.7	-	-
NP	Sentence	Noisy Cloze	×	19.5	29.3	16.6	25.7	-	-
NE	Sub-clause	Identity	✓	24.2	34.6	12.6	21.5	-	-
NE	Sub-clause	Identity	×	21.9	31.9	16.1	26.8	-	-
NE	Sentence	Identity	×	18.1	27.4	12.4	21.2	-	-
NP	Sentence	Identity	×	14.6	23.9	6.6	13.5	-	-
NE	Rule-Based (Heilman and Smith, 2010)			28.2	41.5	23.1	40.2	-	-
NP	Rule-Based (Heilman and Smith, 2010)			16.0	37.9	13.8	35.4	-	-

Table 3.3: Ablations on the SQuAD development set. "Wh* Heuristic" indicates if a heuristic was used to choose sensible wh*-words during cloze translation. NE and NP refer to named entity mention and noun phrase answer generation.

port mean results. Results are shown in Table 3.3, and observations and aggregated trends are highlighted below.

Posterior Maximisation vs. Training on Generated Data Comparing Posterior Maximisation with BERT-Base and BiDAF+SA columns in Table 3.3 shows that training RC models is more effective than maximising question likelihood. As shown later, this could partly be attributed to RC models being able to generalise answer spans, returning answers at test-time that are not always named entity mentions. The RC method, when using BERT, is also able to draw on more powerful pretrained representations.

Effect of Answer Prior Named Entities (NEs) are a more effective answer prior than noun phrases (NPs). Equivalent BERT-Base models trained with NEs improve on average by 8.9 F1 over NPs. Rajpurkar et al. (2016) estimate 52.4% of answers in SQuAD are NEs, whereas (assuming NEs are a subset of NPs), 84.2% are NPs. However, we found that there are on average 14 NEs per context compared to 33 NPs, so using NEs in training may help reduce the search space of possible answer candidates a model must consider.

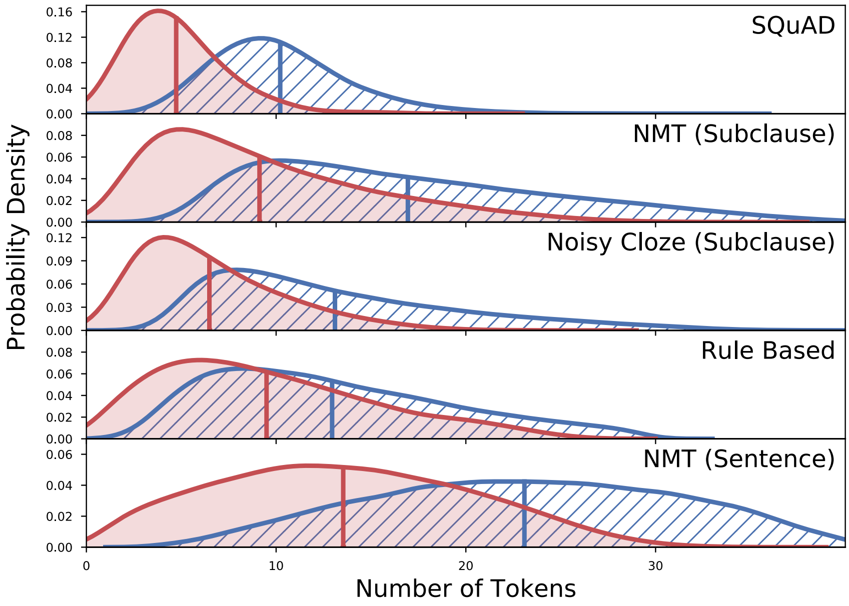


Figure 3.2: Lengths (blue, hashed) and longest common sub-sequence with context (red, solid) for SQuAD questions and various question generation methods.

Effect of Question Length and Lexical Overlap As shown in Figure 3.2, using sub-clauses for generation leads to shorter questions and shorter common sub-sequences to the context, which more closely match the distribution of SQuAD questions. Reducing the length of cloze questions helps the translation components produce simpler, more precise questions. Using sub-clauses leads to, on average +4.0 F1 across equivalent sentence-level BERT-Base models.

Effect of Cloze Translation Noise acts as helpful regularisation when comparing the “identity” cloze translation functions to “noisy cloze”, (+9.8 F1 across equivalent BERT-Base models). UMT question translation is also helpful, leading to an additional mean improvement of 1.8 F1 on BERT-Base for otherwise equivalent “noisy cloze” models. The improvement from UMT is most pronounced when the wh* heuristic is not used, (+4.2 F1 for BERT). However, in general, noisy clozes are surprisingly effective baselines, which we shall discuss further in Section 3.4.

Effect of RC model BERT-Base is more effective than BiDAF+SA (an architecture specifically designed for RC). BERT-Large gives a further boost, improving our best configuration by 6.9 F1.

Question Generation	EM	F1
Rule-Based (NE Answers)	28.2	41.5
Ours	38.6	47.8
Ours (filtered for *c*,*a* pairs in Rule-Based)	38.5	44.7

Table 3.4: SQuAD development set ablations investigating rule-based system accuracy.

Effect of Rule-based Generation RC models trained on questions generated by the Rule-based model of Heilman and Smith (2010) do not perform favourably compared to our MT approach. We note that the rule-based system generates questions from comparatively few c,a pairs, due to its restricted set of rule templates. This means there is less variety in the rule-based generator's RC training data. To measure whether this is significant source of lost accuracy, we remove c,a pairs from our best-preforming synthetic training set that the rule-based system cannot generate questions from. Table 3.4 shows that this hurts our model's performance, indicating that variety in c,a pairs is important, but it does not fully explain the difference. Also, whilst on average, question lengths are shorter for the rule-based model than the UMT model, the distribution of longest common sequences are similar, as shown in Figure 3.2, perhaps suggesting that the RB system copies a larger proportion of its input.

3.3.3 Error Analysis

We find that the BERT RC model predicts answer spans that are not always detected as named entity mentions (NEs) by the NER tagger, despite being trained with solely NE answer spans. In fact, when we split SQuAD into questions where the correct answer is an automatically tagged NE, our model's performance improves to 64.5 F1, but it still achieves 47.9 F1 on questions which do not have automatically tagged NE answers (not shown in our tables). We attribute this to the effect of BERT's linguistic pretraining allowing it to generalise the semantic role played by NEs rather than simply learning to mimic the NER system. Indeed, an equivalent BiDAF+SA model scores 58.9 F1 when the answer is an NE but drops severely to 23.0 F1 when the answer is not an NE.

Figure 3.3 shows the performance of our best RC model for different kinds of ques-

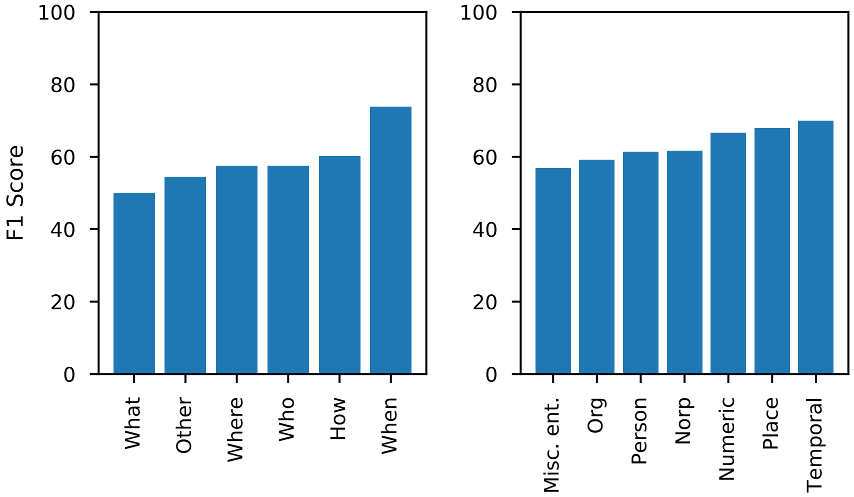


Figure 3.3: Breakdown of performance for our best RC model on SQuAD for different question types (left) and different NE answer categories (right)

tion and answer type. The model performs best with "when" questions which tend to have fewer potential answers to choose between, but struggles with "what" questions, which have a broader range of answer semantic types, and hence more plausible answers per context. The model performs well on "TEMPORAL" answers, consistent with the good performance of "when" questions.

3.3.4 UMT-generated Question Analysis

Whilst our main aim is to optimise for downstream RC performance, it is also instructive to examine the output of the unsupervised MT cloze translation system. Unsupervised MT has been used in monolingual settings in previous work (Subramanian et al., 2018), but cloze-to-question generation presents new challenges – The cloze and question domains are asymmetric in terms of word length, and successful translation must preserve the answer, not just superficially transfer style. Figure 3.4 shows that without the wh* heuristic, the UMT model learns to generate questions with broadly appropriate wh*-words for the answer type, but can struggle, particularly with PERSON/ORG/NORP and NUMERIC answers.

Table 3.6 shows representative examples from the NE UMT model. The model generally copies large segments of the input. Also shown in Figure 3.2, generated questions have, on average, a 9.1 token contiguous sub-sequence from the context, corresponding to 56.9% of a generated question copied verbatim, compared to 4.7 tokens (46.1%) for SQuAD questions. This is unsurprising, as the back-translation training objective is to maximise the reconstruction of inputs, which encourages conservative translation.

Clearly, there is much room for improvement. However, the model exhibits some encouraging, non-trivial syntax manipulation and generation, particularly at the start of questions, such as example 7 in Table 3.6, where word order is significantly modified and “sold” is replaced by “buy”. Occasionally, the model hallucinates common patterns in the question corpus (example 6). It can struggle with lists (example 4), and often prefers present tense and the second person (example 5). Finally, semantic drift is an issue, with generated questions being relatively coherent but often having different answers to the source cloze questions (example 2).

We can estimate how well-formed the questions generated by various configurations of our model are using the Well-formed query dataset of Faruqui and Das (2018). This dataset consists of 25,100 search engine queries, annotated with whether the query is a well-formed question or not. We train a BERT-Base classifier on the binary classification task, achieving a test set accuracy of 80.9%. We then use this classifier to measure what proportion of questions generated by our models are classified as “well-formed”. Table 3.5 shows the full results. Our best unsupervised question generation configuration generates 68.0% well-formed questions. The rule-based generator achieves 75.6%, consistent with our observations that the rule-based model produces more grammatically accurate questions. The classifier predicts that 92.3% of SQuAD questions are well-formed, suggesting it is able to detect high quality questions. The well-formedness classifier appears to be sensitive to fluency and grammar, with the “identity” cloze translation models scoring much higher than their “noisy cloze” counterparts.

3.3.5 Few-Shot Question Answering

Finally, we consider a few-shot learning task with very limited training examples. We follow the methodology of Dhingra et al. (2018) and Yang et al. (2017), training

Cloze Answer	Cloze Boundary	Cloze Translation	Wh* Heuristic	% Well-formed
NE	Sub-clause	UMT	✓	68.0
NE	Sub-clause	UMT	×	65.3
NE	Sentence	UMT	×	61.3
NP	Sentence	UMT	×	61.9
NE	Sub-clause	Noisy Cloze	✓	2.7
NE	Sub-clause	Noisy Cloze	×	2.4
NE	Sentence	Noisy Cloze	×	0.7
NP	Sentence	Noisy Cloze	×	0.8
NE	Sub-clause	Identity	✓	30.8
NE	Sub-clause	Identity	×	20.0
NE	Sentence	Identity	×	49.5
NP	Sentence	Identity	×	48.0
Rule-Based (Heilman and Smith, 2010)				75.6
SQuAD Questions (Rajpurkar et al., 2016)				***92.3***

Table 3.5: Fraction of questions classified as "well-formed" by a classifier trained on the dataset of Faruqui and Das (2018) for different question generation models.

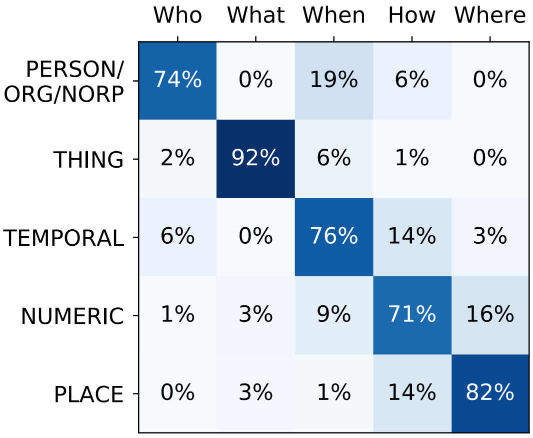


Figure 3.4: wh*-words generated by the UMT model for cloze questions with different answer types.

on a small number of training examples and using the development set for early stopping. We first train a BERT-large RC model using our best configuration from Section 3.3, then fine-tune with the small amount of SQuAD training data. We compare to our re-implementation of Dhingra et al. (2018), and training the RC model directly on the training data without unsupervised RC training.

Figure 3.5 shows performance for progressively larger amounts of training data. As with Dhingra et al. (2018), our numbers are attained using a development set for

#	Cloze Question	Answer	Generated Question
1	they joined with [PERSON/NORP/ORG] to defeat him	Rom	Who did they join with to defeat him?
2	the [NUMERIC] on Orchard Street remained open until 2009	second	How much longer did Orchard Street remain open until 2009?
3	making it the 3rd largest football ground in [PLACE]	Portugal	Where is it 3rd the third football ground?
4	he speaks [THING], English, and German	Spanish	What are we , English , and German?
5	Arriving in the colony early in [TEMPORAL]	1883	When are you in the colony early?
6	The average household size was [NUMERIC]	2.30	How much does a Environmental Engineering Technician II in Suffolk , CA make?
7	WALA would be sold to the Des Moines-based [PERSON/NORP/ORG] for $86 million	Meredith Corp	Who would buy the WALA Des Moines-based for $86 million?

Table 3.6: Examples of cloze translations for the UMT model using the wh* heuristic and sub-clause cloze extraction. More examples can be found in appendix C.5

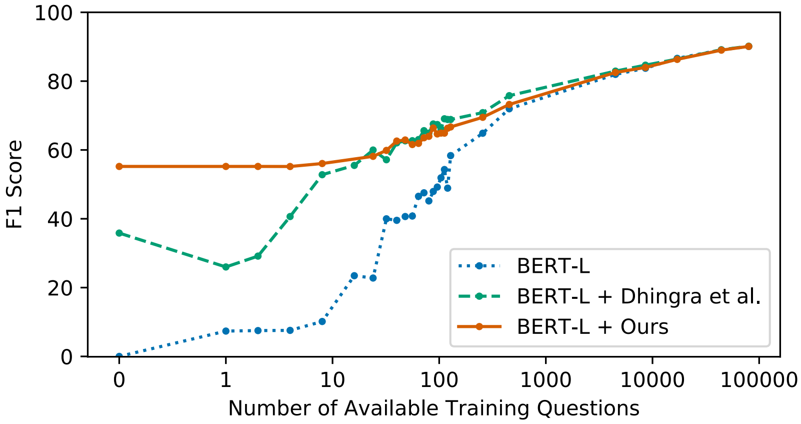


Figure 3.5: F1 score on the SQuAD development set for progressively larger training dataset sizes

early stopping that can be larger than the training set. Hence this is not a true reflection of performance in low data regimes, but does allow for comparative analysis between models (Perez et al., 2021). We find our approach performs best in very data poor regimes, and similarly to Dhingra et al. (2018) with modest amounts of data. We also note BERT-Large itself, without pretraining, is surprisingly efficient, reaching ∼60 F1 with only 1% of the training set (1000 examples). As a comparison, BiDAF+SA needs 20K examples to reach 60 F1 (not shown on figure).

3.4 Reflection

It is worth noting that to attain our best performance, we require the use of both an NER system, indirectly using labelled data from OntoNotes 5, and a constituency parser for extracting sub-clauses, trained on the Penn Treebank (Marcus et al., 1994).[3] Moreover, a language-specific wh* heuristic was used for training the best-performing MT models. This limits the applicability and flexibility of our best-performing approach to domains and languages that already enjoy extensive linguistic resources (named entity recognition and treebank datasets), as well as requiring some human engineering to define new heuristics.

Nevertheless, our approach is unsupervised from the perspective of requiring no labelled q,a or q,c pairs let alone any aligned q,c,a triples, which are usually the most challenging aspects of annotating RC training datasets.

We note the "noisy cloze" system, consisting of very simple rules and noise, performs nearly as well as our more complex best-performing system, despite the lack of grammaticality and syntax associated with questions. The questions generated by the noisy cloze system also perform poorly on the "well-formedness" analysis mentioned in Section 3.3.4, with only 2.7% classified as well-formed. This intriguing result suggests natural questions are perhaps less important for SQuAD and strong question-context word matching is enough to do well, reflecting work from Jia and Liang (2017) who demonstrate that even supervised models rely on word-matching. A subsequent evaluation of our model on the ADDSENT adversarial test set of Jia and Liang (2017) saw scores drop from 56% to 29%, demonstrating its brittleness, and lending evidence to this interpretation.

Additionally, questions generated by our approach require no multi-hop or multi-sentence reasoning, but can still be used to achieve non-trivial SQuAD performance. Indeed, Min et al. (2018) note 90% of SQuAD questions only require a single sentence of context, and Sugawara et al. (2018) find 76% of SQuAD has the answer in the sentence with highest token overlap to the question.

3.5 Related Work

Here we shall highlight some related work of specific interest to this chapter, which has not been previously discussed in chapter 2.

[3] Ontonotes 5: https://catalog.ldc.upenn.edu/LDC2013T19

Unsupervised Learning in NLP Most representation learning approaches use latent variables (Hofmann, 1999; Blei et al., 2003), or language model-inspired criteria (Collobert and Weston, 2008; Mikolov et al., 2013; Pennington et al., 2014; Radford et al., 2018; Devlin et al., 2019). Most relevant to us is unsupervised MT (Lample et al., 2018b,a,c; Artetxe et al., 2018) and style transfer (Subramanian et al., 2018). We build upon this work, but instead of using models directly, we use them for training data generators. Yadav et al. (2019) propose an unsupervised alignment method for multiple choice QA. Radford et al. (2019) report that powerful language models such as GPT2 can be used to perform a number of unsupervised tasks. Their method, which differs substantially to ours, revolves around prompting. For example, they demonstrate unsupervised summarisation of long paragraphs by prompting the Language model to generate a summary with the prompt "TL;DR". More relevant to our setting, they demonstrate that this approach can answer questions from a conversational RC task, CoQA (Reddy et al., 2019) in an unsupervised manner, using the conversation history of question-answer pairs as a prompt. Chan et al. (2019) report unsupervised SQuAD results for GPT2 of 17%, and introduce a model called KERMIT, which improves to 30%, bot well below our result of 56%. Moreover, both of these models assume SQuAD questions in cloze-format rather than the harder natural question format we consider. GPT3 (Brown et al., 2020) builds on prompting formula from Radford et al. (2019), and, by using an extremely large 175 billion parameter model, demonstrates prompted unsupervised SQuAD results of 59%, outperforming our approach by 4%, but this requires a model 500× the size. Moreover, the prompting approach is orthogonal to ours, and recent best practice recommends a combination of prompting and fine-tuning for low-data settings (Le Scao and Rush, 2021).

Semi-supervised RC Yang et al. (2017) train a RC model and also generate new questions for greater data efficiency, but require labelled data. Dhingra et al. (2018) simplify the approach and remove the supervised requirement for question generation, but do not target unsupervised RC or attempt to generate natural questions. They also make stronger assumptions about the text used for question generation and require Wikipedia summary paragraphs. Wang et al. (2018b) consider semi-supervised cloze RC, Chen et al. (2018) use semi-supervision to improve semantic parsing on WebQuestions (Berant et al., 2013), and Lei et al. (2016) leverage semi-supervision for question similarity modelling. Golub et al. (2017) propose a method to generate domain specific training RC instances for transfer learning between SQuAD and NewsQA (Yadav et al., 2019). Finally, injecting external knowledge into QA systems could be viewed as semi-supervision, and Weissenborn et al. (2017a) and Mihaylov and Frank (2018) use ConceptNet (Speer et al., 2016) for QA tasks. Recently, after the work in this chapter was performed, there has been an explosion of interest in few-shot learning, inspired by the success of GPT3 (Schick and Schütze, 2021a,b; Gao et al., 2021b; Le Scao and Rush, 2021; Tam et al., 2021). Many of these recently developed techniques are orthogonal to our generation approach, and could be combined to improve the learning curves in Figure 3.5. Finally, Ram et al. (2021) report impressive recent results on few-shot RC, achieving similar results as in figure 3.5 with a different mechanism.

Question Generation has been tackled with pipelines of templates and syntax rules (Rus et al., 2010). Heilman and Smith (2010) augment this with a model to rank generated questions, and Yao et al. (2012) and Olney et al. (2012) investigate symbolic approaches. There has been interest in question generation using supervised neural models, many trained to generate questions from c,a pairs in SQuAD (Du et al., 2017; Yuan et al., 2017; Zhao et al., 2018; Du and Cardie, 2018; Hosking and Riedel, 2019). Some of these supervised generators can generate RC data of sufficient quality they can be used to train RC models equal to those trained on real RC data (Alberti et al., 2019; Puri et al., 2020). Since the experiments in this chapter were performed, natural language generation has improved greatly, with the introduction of powerful generative transformers, such as GPT2/3, T5 (Raffel et al., 2020) and BART (Lewis et al., 2020a). These could be used to initialize $p_{s \to t}$ and $p_{t \to s}$, which would likely substantially improve generation quality.

3.6 Conclusion

In this chapter, we set out to explore the span-extraction RC task in the absence of annotated RC training data. Our motivations were to assess the empirical feasibility of unsupervised RC, in order to understand to what extent annotated data is really necessary, and to provide a good starting point for few-shot RC.

We have found that by using a synthetic RC dataset generation approach, leveraging a careful design and recent advances in UMT, coupled with powerful pretrained RC models, was sufficient to surpass some simple fully supervised systems. The synthetic dataset generation paradigm is attractive as it allows us to flexibly incorporate priors into models (such as encouraging named-entity-like answers). It also decouples generation from the RC model training, allowing for advances in RC models to be leveraged as and when they are developed in future work.

However, we note that whilst our results are encouraging on the relatively simple SQuAD task, our method relies on access to linguistic resources and heuristics. Specifically, for our best result, a heuristic was used to help our generator learn the connection between answer types and wh*-words, and a syntax parser and Named Entity Recogniser was required. In general, we may not be able to assume these exist in a language or domain of interest, which limits the applicability of our method. That said, such resources do tend to be more available than RC annotated data. In the next chapter, we shall explore a different approach to tackling RC in languages which do not have in-language RC annotations.

The RC behaviour induced in our model, whilst non-trivial, is brittle. Through thorough analysis, we discovered that our model is able to perform some limited generalisation from its answer prior. However, on the balance of the available evidence, its behaviour is best described as mostly simple candidate answer location based on recognising simple patterns from questions, and disambiguation between candidate answers by a soft, noise-resistant lexical phrase matching. Despite its simplicity, this behaviour serves as good foundation for further fine-tuning with real annotated data, as shown by our few-shot results.

We have limited our study here to RC. This approach could be extended to ODQA, by adding an unsupervised retriever, such as TF-IDF, and then using the unsupervised RC model to produce a final answer – effectively an unsupervised analogue of retrieve-and-read models like DrQA (Chen et al., 2017). We shall explore a similar approach, capable of unsupervised cloze-question ODQA in chapter 5.

Future work in this area should attempt to handle more challenging RC elements and reduce reliance on linguistic resources and heuristics. Promising innovations in pretrained natural language generators, which we use in later chapters, would also likely improve the quality of generated questions. Additional perspectives and remarks on future work can be found in the conclusion of this thesis, chapter 9.

Chapter 4

Evaluating Cross-Lingual Reading Comprehension

In the previous chapter, we saw how RC behaviour could be induced without needing annotated RC training data. One of our stated motivations for doing so was to assist in domains and languages where there was no RC training data available. Indeed, RC datasets (and QA dataset in general) in languages other than English remain scarce, even for relatively high-resource languages (Asai et al., 2018), since collecting such datasets at sufficient scale and quality is difficult and costly.

In this chapter, we shall more directly address the problem of multi-lingual RC. In general, we cannot assume there will be any training RC data available in our target language. We could apply the method from chapter 3 directly in the language of interest. However, this would have two drawbacks. First, as we highlighted at the end of chapter 3, this would require access to linguistic resources, such as syntax parsers and noun chunkers, or NER systems, which might not be available. Second, this approach would not make use of existing RC training data that may be available, albeit not in the language of interest.

There is, however, a more fundamental barrier to the development of multilingual QA systems that must be overcome. Since there we do not have any annotated RC data in most languages, including evaluation data, we cannot even *measure* progress on multilingual QA. Given recent progress in cross-lingual tasks such as document classification (Lewis et al., 2004; Klementiev et al., 2012; Schwenk and Li, 2018), semantic role labelling (Akbik et al., 2015) and NLI (Conneau et al., 2018), we argue that whilst multilingual RC training data might be useful but not strictly necessary, multilingual *evaluation data* is a must-have.

Recognising this need, several cross-lingual RC datasets have recently been assembled (Asai et al., 2018; Liu et al., 2019a). However, these generally cover only a small number of languages, combine data from different authors and annotation protocols, lack parallel instances, or explore less practically useful settings or tasks. Highly parallel data is particularly attractive, as it enables fairer comparison across languages, requires fewer source language annotations, and allows for additional evaluation setups at no extra annotation cost. A purpose-built evaluation benchmark dataset covering a range of diverse languages, and following the popular span-extractive RC paradigm on a practically useful domain would be a powerful testbed for cross-lingual RC. In this chapter, we construct such a dataset, which we refer to as MLQA, in order to evaluate and accelerate progress on multilingual RC.

Bigger Picture As mentioned above, this chapter builds on the low data themes from the first chapter, but focuses on a more practical, less artificial setting. The focus is still on RC, whereas ultimately, we want to tackle the more ODQA task – see the "bigger picture" paragraph in the introduction of chapter 3 for additional commentary. Following the this chapter, Chapter 5 will continue to examine how to perform QA without annotated data, but will shift the focus to ODQA. This chapter primarily revolves around the challenges, and techniques and solutions to creating datasets to measure multilingual transfer in QA. The theme of test-time measurement of QA behaviours will also be key features of chapters 5 and 7, where the focus is not on training models, but analysing what kinds of behaviours models exhibit. The human-annotation procedure used in this chapter will inspire a similar human-labour saving technique for annotation in chapter 7, which will also use a retrieval-like technique to select instances for human annotation. Additional commentary on the connections between this chapter and the wider body of work in the thesis can be found in the conclusion of this chapter (sec. 4.7) and the thesis conclusion, chapter 9.

The material in this chapter first appeared in:

> **Patrick Lewis**, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020b. MLQA: Evaluating Cross-lingual Extractive Question Answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*
>
> *Individual Contributions: The initial idea was proposed by the thesis author. The development of the dataset construction methodology was a close collaboration between the thesis author and a co-author. Prepossessing, English QA annotation, analysis and the majority of the writing of the original paper was carried out by the thesis author. Parallel sentence mining, RC Modelling experiments and non-English annotation were performed by co-authors.*

4.1 Overview

MLQA is multi-way parallel across seven languages: *English, Arabic, German, Vietnamese, Spanish, Simplified Chinese* and *Hindi*. To construct MLQA, we first automatically identify sentences from Wikipedia articles which have the same or similar meaning in multiple languages using a multilingual dense retrieval technique. We extract the paragraphs that contain such sentences, then crowd-source questions on the English paragraphs, making sure the answer is in the aligned sentence. This makes it possible to answer the question in all languages in the vast majority of cases.[1] Professional translators then translate the questions into all target languages, and annotate answer spans in the aligned contexts.

The resulting dataset has between 5,000 and 6,000 instances in each language, and more than 12,000 in English. Each instance has an aligned equivalent in multiple other languages (always including English), the majority being 4-way aligned. Combined, there are over 46,000 RC annotations.

[1] The automatically aligned sentences occasionally differ in a named entity or information content. In these rare cases, there may be no answer for some languages.

We define two tasks to assess multilingual RC performance on MLQA. The first, cross-lingual transfer (XLT), requires models trained in one language (in our case English) to transfer to test data in a different language, without using any annotated RC data in the test language. The second, generalised cross-lingual transfer (G-XLT) requires models to answer questions where the question and context languages are *different*, e.g. questions in Hindi and contexts in Arabic, a setting only possible since MLQA is highly parallel.

We evaluate a number of modelling approaches for these tasks. We develop machine translation baselines which map answer spans based on the attention matrices from a translation model, and use multilingual BERT (M-BERT, Devlin et al., 2019) and XLM (Conneau and Lample, 2019) as zero-shot approaches. We use English for our training language and adopt SQuAD as a training dataset. We find that XLM transfers best, but all models lag well behind training-language performance.

In summary, we make the following contributions: i) We develop a novel annotation pipeline to construct large multilingual, highly parallel RC datasets ii) We release MLQA, a 7-language evaluation dataset for RC iii) We define two cross-lingual RC tasks, including a novel generalised cross-lingual RC task iv) We evaluate a series of modelling approaches, and find that cross-lingual representation models such as XLM surpass translation-based approaches.

4.2 The MLQA Dataset

First, we state our desired properties for a cross-lingual RC evaluation dataset. We then describe our annotation protocol, which seeks to fulfil these desiderata.

Parallel The dataset should consist of instances that are parallel across many languages. First, this makes comparison of RC accuracy as a function of transfer language fairer. Second, additional evaluation setups become possible, as questions in one language can be applied to documents in another. Finally, annotation cost is also reduced as more instances can be shared between languages.

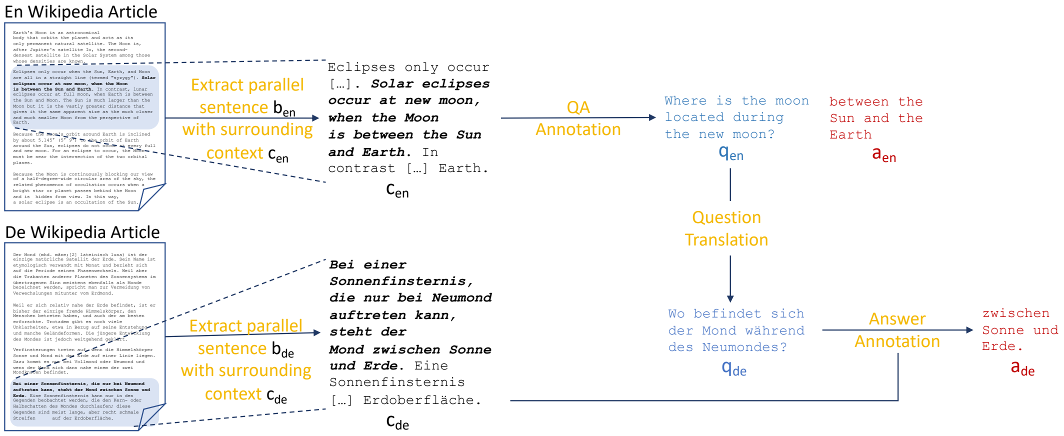

En Wikipedia Article
Extract parallel sentence b_en with surrounding context c_en
Eclipses only occur […]. Solar eclipses occur at new moon, when the Moon is between the Sun and Earth. In contrast […] Earth.
c_en
QA Annotation
Where is the moon located during the new moon?
q_en
between the Sun and the Earth
a_en
Question Translation
De Wikipedia Article
Extract parallel sentence b_de with surrounding context c_de
Bei einer Sonnenfinsternis, die nur bei Neumond auftreten kann, steht der Mond zwischen Sonne und Erde. Eine Sonnenfinsternis […] Erdoberfläche.
c_de
Wo befindet sich der Mond während des Neumondes?
q_de
Answer Annotation
zwischen Sonne und Erde.
a_de

Figure 4.1: MLQA annotation pipeline. Only one target language is shown for clarity. Left: We first identify N-way parallel sentences $b_{en}, b_{\text{lang }1} \ldots b_{\text{lang }N-1}$ in Wikipedia articles on the same topic, and extract the paragraphs that contain them, c_{en}, $c_{\text{lang }1} \ldots c_{\text{lang }N-1}$. Middle: Workers formulate questions q_{en} from c_{en} for which answer a_{en} is a span within b_{en}. Right: English questions q_{en} are then translated by professional translators into all languages to obtain $q_{\text{lang }1} \ldots q_{\text{lang }N-1}$. Finally, the answer is annotated in the target language contexts by professional translators, such that $a_{\text{lang }i}$ is a span within $b_{\text{lang }i}$.

Natural Documents Building a parallel RC dataset in many languages requires access to parallel documents in those languages. Manually translating documents at sufficient scale entails very large translator workloads, and could result in unnatural documents, due the phenomenon of "Translationese": the tendency for manual translation to introduce artefacts leading to unrepresentative text (Lembersky et al., 2011; Volansky et al., 2015). Exploiting existing naturally parallel texts is attractive, providing high-quality documents without requiring manual translation.

Diverse Languages A primary goal of cross-lingual research is to develop systems that work well in many languages. The dataset should enable quantitative performance comparison across languages with different linguistic resources, language families and scripts.

Textual Domain We require a naturally highly language-parallel textual domain. It is also desirable to select a textual domain that matches existing RC training resources, in order to isolate changes in performance due to language transfer.

To satisfy these desiderata, we designed the method described below and illustrated in Figure 4.1. Wikipedia represents a convenient textual domain, as its size and multilinguality enables collection of data in many diverse languages at scale. We choose English as our source language as it has the largest Wikipedia. We choose six other languages which exhibit a broad range of linguistic phenomena and have sufficiently large Wikipedias. Our annotation pipeline consists of three main steps:

1. We automatically extract paragraphs which contain a parallel sentence from articles on the same topic in each language (left of Figure 4.1).

2. We employ crowd-workers to annotate questions and answer spans on the English paragraphs (centre of Figure 4.1). Annotators must choose answer spans within the parallel source sentence. This allows annotation of questions in the source language with high probability of being answerable in the target languages, even if the rest of the context paragraphs are different.

3. We employ professional translators to translate the questions and to annotate answer spans in the target language (right of Figure 4.1).

The following describes each step in the collection pipeline in more detail.

4.2.1 Parallel Sentence Mining

Parallel Sentence mining allows us to leverage naturally written documents and avoid translation, which would be expensive and result in potentially unnatural documents. In order for questions to be answerable in every target language, we use contexts containing an N-way parallel sentence. Our approach is similar to WikiMatrix (Schwenk et al., 2021) which extracts parallel sentences for many language pairs in Wikipedia, but we limit the search for parallel sentences to documents on the same article only, and aim for N-way parallel sentences.

To detect parallel sentences we use LASER[2] a powerful open-source tool for parallel sentence mining employing dense retrieval (Artetxe and Schwenk, 2019a). LASER uses multilingual sentence embeddings and a distance in the embeddings space to detect parallel sentences. The reader is referred to Artetxe and Schwenk

[2] https://github.com/facebookresearch/LASER

de	es	ar	zh	vi	hi
5.4M	1.1M	83.7k	24.1K	9.2k	1340

Table 4.1: Incremental alignment with English to obtain 7-way aligned sentences.

(2019b) and Artetxe and Schwenk (2019a) for a detailed description.

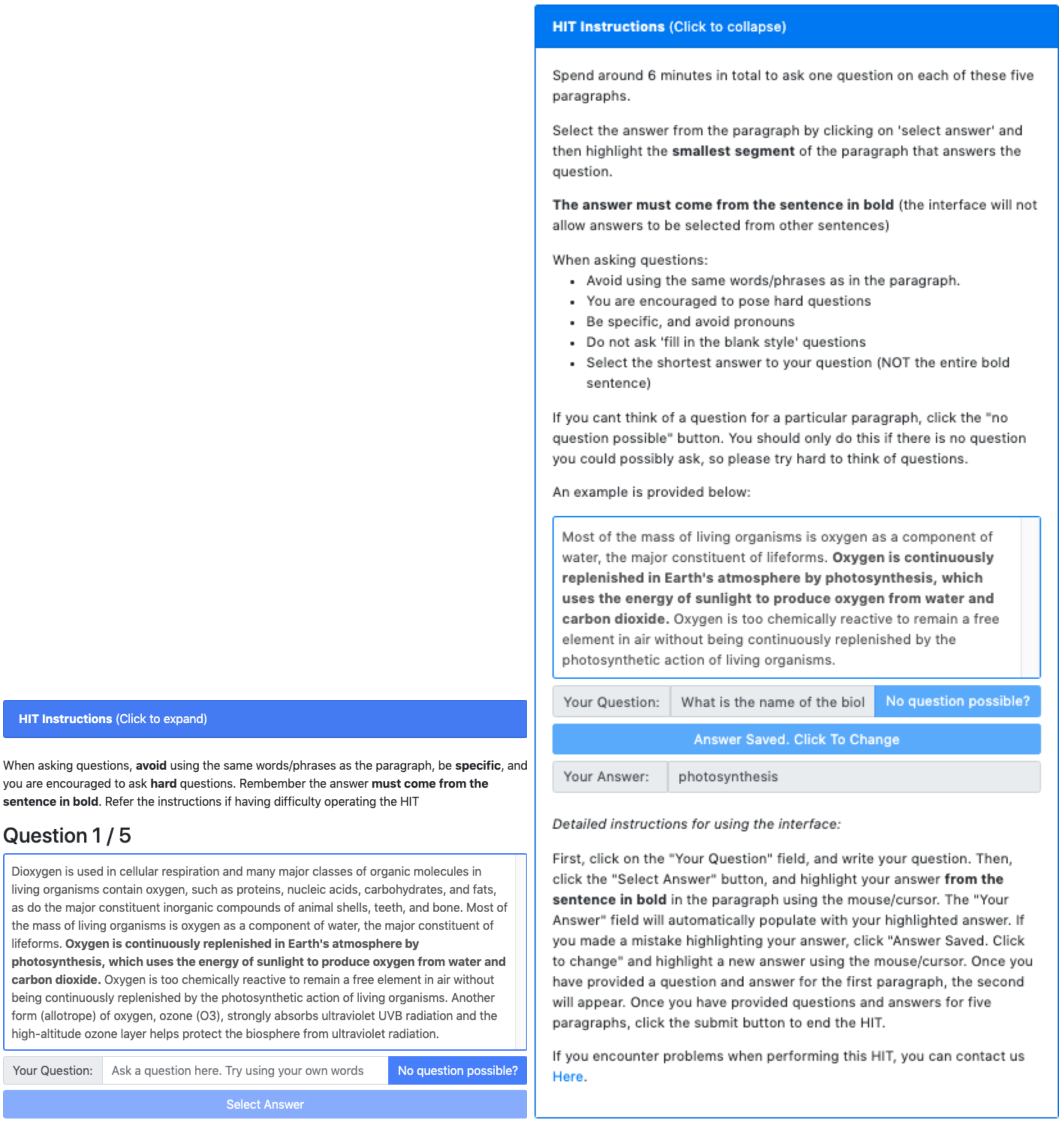
We begin with the May 2019 Wikipedia dump for all our target languages, which we preprocess using WikiExtractor (Attardi, 2015), with a number of modifications and additional cleanup steps to support the idiosyncrasies of each language's dump. Additionally, OpenCC[3] is used to convert all Chinese contexts to Simplified Chinese, as Chinese Wikipedia dumps generally consist of a mixture of simplified and traditional Chinese text. We then independently align all languages with English, then intersect these sets of parallel sentences, forming sets of N-way parallel sentences. As shown in Table 4.1, starting with 5.4M parallel English/German sentences, the number of N-way parallel sentences quickly decreases N increases, i.e. as more languages are added. We also found that 7-way parallel sentences lack linguistic diversity, and often appear in the first sentence or paragraph of articles.

As a compromise between language-parallelism and both the number and diversity of parallel sentences, we use sentences that are 4-way parallel. This yields 385,396 parallel sentences (see Appendix D.1 for exhaustive details) which were sub-sampled to ensure parallel sentences were evenly distributed in paragraphs. We ensure that each language combination is equally represented, so that each language will share many parallel RC instances with every other language. Except for any rejected instances later in the pipeline, each RC instance will be parallel between English and three target languages.

4.2.2 English RC Annotation

We use Amazon Mechanical Turk to annotate English RC instances, broadly following the SQuAD methodology (Rajpurkar et al., 2016). We present workers with an English aligned sentence, b_{en} along with the paragraph that contains it c_{en}. Work-

[3] https://github.com/BYVoid/OpenCC



When asking questions, **avoid** using the same words/phrases as the paragraph, be **specific**, and you are encouraged to ask **hard** questions. Rembember the answer **must come from the sentence in bold**. Refer the instructions if having difficulty operating the HIT

Question 1 / 5

Dioxygen is used in cellular respiration and many major classes of organic molecules in living organisms contain oxygen, such as proteins, nucleic acids, carbohydrates, and fats, as do the major constituent inorganic compounds of animal shells, teeth, and bone. Most of the mass of living organisms is oxygen as a component of water, the major constituent of lifeforms. **Oxygen is continuously replenished in Earth's atmosphere by photosynthesis, which uses the energy of sunlight to produce oxygen from water and carbon dioxide.** Oxygen is too chemically reactive to remain a free element in air without being continuously replenished by the photosynthetic action of living organisms. Another form (allotrope) of oxygen, ozone (O3), strongly absorbs ultraviolet UVB radiation and the high-altitude ozone layer helps protect the biosphere from ultraviolet radiation.

Your Question: Ask a question here. Try using your own words

Spend around 6 minutes in total to ask one question on each of these five paragraphs.

Select the answer from the paragraph by clicking on 'select answer' and then highlight the **smallest segment** of the paragraph that answers the question.

The answer must come from the sentence in bold (the interface will not allow answers to be selected from other sentences)

When asking questions:

- Avoid using the same words/phrases as in the paragraph.
- You are encouraged to pose hard questions
- Be specific, and avoid pronouns
- Do not ask 'fill in the blank style' questions
- Select the shortest answer to your question (NOT the entire bold sentence)

If you cant think of a question for a particular paragraph, click the "no question possible" button. You should only do this if there is no question you could possibly ask, so please try hard to think of questions.

An example is provided below:

Most of the mass of living organisms is oxygen as a component of water, the major constituent of lifeforms. **Oxygen is continuously replenished in Earth's atmosphere by photosynthesis, which uses the energy of sunlight to produce oxygen from water and carbon dioxide.** Oxygen is too chemically reactive to remain a free element in air without being continuously replenished by the photosynthetic action of living organisms.

Your Question: What is the name of the biol

Your Answer: photosynthesis

Detailed instructions for using the interface:

First, click on the "Your Question" field, and write your question. Then, click the "Select Answer" button, and highlight your answer **from the sentence in bold** in the paragraph using the mouse/cursor. The "Your Answer" field will automatically populate with your highlighted answer. If you made a mistake highlighting your answer, click "Answer Saved. Click to change" and highlight a new answer using the mouse/cursor. Once you have provided a question and answer for the first paragraph, the second will appear. Once you have provided questions and answers for five paragraphs, click the submit button to end the HIT.

If you encounter problems when performing this HIT, you can contact us Here.

Figure 4.2: English annotation interface (left) and detailed annotations instructions (right)

ers formulate a question q_{en} and highlight the shortest answer span a_{en} that answers it. a_{en} must be a span within b_{en} to ensure q_{en} will be answerable in the target languages. We include a "No Question Possible" button when no sensible question could be asked. Figure 4.2 shows a screenshot of the annotation interface. There are a number of data input validation features to assist workers, as well as detailed instructions in a drop-down window.

The first 15 questions from each worker are manually checked, after which, if their work was satisfactory, their future work is auto-approved. Otherwise, the worker is contacted with feedback with how to improve, and the process is repeated.

Once the questions and answers have been annotated, we run another task to re-annotate English answers. Here, workers are presented with q_{en} and c_{en}, and requested to generate an a'_{en} or to indicate that q_{en} is not answerable. Two additional answer span annotations are collected for each question. The additional answer annotations enable us to calculate an inter-annotator agreement (IAA) score. We calculate the mean token F1 score between the three answer annotations, giving an IAA score of 82%, comparable to SQuAD, where this IAA measure is 84%.

Rather than provide all three answer annotations as gold answers, we select a single representative reference answer. In 88% of cases, either two or three of the answers exactly matched, so the majority answer is selected. In the remaining cases, the answer with highest F1 overlap with the other two is chosen. This results both in an accurate answer span, and ensures the English results are comparable to those in the target languages, where only one answer is annotated per question.

We discard instances where annotators marked the question as unanswerable as well as instances where over 50% of the question appeared as a sub-sequence of the aligned sentence, as these are too easy or of low quality. Finally, we reject questions where the IAA score was very low (< 0.3) removing a small number of low quality instances. To verify we were not discarding challenging but high quality examples in this step, a manual analysis of discarded questions was performed. Of these discarded questions, 38% were poorly specified, 24% did not make sense, 30% had poor answers, and only 8% were high-quality challenging questions.

4.2.3 Target Language RC Annotation

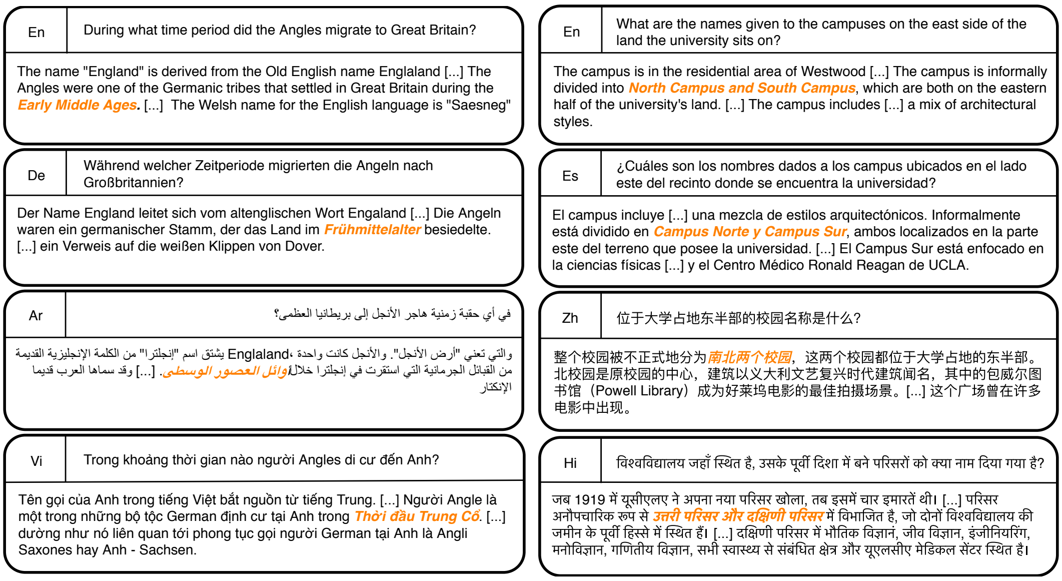
We use the One Hour Translation platform to source professional translators to translate the questions from English to the six target languages, and to find answers in the target contexts. We present each translator with the English question q_{en}, English answer a_{en}, and the context c_x (containing aligned sentence b_x) in target language x. The translators are only shown the aligned sentence and the sentence on each side (where these exist). This increases the chance of the question being answerable, as in some cases aligned sentences are not perfectly parallel, without requiring workers to read the entire context. By providing English answers, we try to minimise cultural and personal differences in the amount of answer detail.

We sample 2% of the translated questions for additional review by language experts. Translators that did not meet the quality standards were removed from the translator pool, and their translations were reallocated. By comparing the distribution of answer lengths relative to the context to the English distribution, some cases were found where some annotators selected very long answers, especially for Chinese. We clarified the instructions with these specific annotators, and send such cases for re-annotation. We discard instances in target languages where annotators indicate there is no answer in that language. This means a small number of instances are no longer 4-way parallel. “No Answer” annotations occurred for 6.6%-21.9% of instances (Vietnamese and German, respectively).

4.2.4 The Resulting MLQA Dataset

Contexts, questions and answer spans for all the languages are then brought together to create the final dataset. MLQA consists of 12,738 extractive RC instances in English and between 5,029 and 6,006 instances in the target languages. 9,019 instances are 4-way parallel, 2,930 are 3-way parallel and 789 2-way parallel. Representative examples are shown in Figure 4.3. MLQA is split into development and test splits, with instance statistics in Tables 4.2a and 4.2b.

Table 4.3 shows the number of Wikipedia articles that feature at least one of their paragraphs as a context paragraph in MLQA, along with the number of unique context paragraphs in MLQA. There are 1.9 context paragraphs from each article on average. This is in contrast to SQuAD, which instead features a small number of curated articles, but is more densely annotated, with 43 context paragraphs per article on average. Thus, MLQA covers a broader range of topics than SQuAD. Figure 4.4 shows the distribution of wh*-words in questions in both MLQA-en and SQuAD v.1. The distributions are very similar, lending evidence that SQuAD is an appropriate surrogate training dataset for MLQA.



En — During what time period did the Angles migrate to Great Britain?

The name "England" is derived from the Old English name Englaland [...] The Angles were one of the Germanic tribes that settled in Great Britain during the ***Early Middle Ages***. [...] The Welsh name for the English language is "Saesneg"

De — Während welcher Zeitperiode migrierten die Angeln nach Großbritannien?

Der Name England leitet sich vom altenglischen Wort Engaland [...] Die Angeln waren ein germanischer Stamm, der das Land im ***Frühmittelalter*** besiedelte. [...] ein Verweis auf die weißen Klippen von Dover.

Ar — في أي حقبة زمنية هاجر الأنجل إلى بريطانيا العظمى؟

والتي تعني "أرض الأنجل". والأنجل كانت واحدة ،Englaland يشتق اسم "إنجلترا" من الكلمة الإنجليزية القديمة من القبائل الجرمانية التي استقرت في إنجلترا خلال***أوائل العصور الوسطى***. [...] وقد سماها العرب قديما الإنكتار

Vi — Trong khoảng thời gian nào người Angles di cư đến Anh?

Tên gọi của Anh trong tiếng Việt bắt nguồn từ tiếng Trung. [...] Người Angle là một trong những bộ tộc German định cư tại Anh trong ***Thời đầu Trung Cổ***. [...] dường như nó liên quan tới phong tục gọi người German tại Anh là Angli Saxones hay Anh - Sachsen.

(a) MLQA example parallel for En-De-Ar-Vi.

En — What are the names given to the campuses on the east side of the land the university sits on?

The campus is in the residential area of Westwood [...] The campus is informally divided into ***North Campus and South Campus***, which are both on the eastern half of the university's land. [...] The campus includes [...] a mix of architectural styles.

Es — ¿Cuáles son los nombres dados a los campus ubicados en el lado este del recinto donde se encuentra la universidad?

El campus incluye [...] una mezcla de estilos arquitectónicos. Informalmente está dividido en ***Campus Norte y Campus Sur***, ambos localizados en la parte este del terreno que posee la universidad. [...] El Campus Sur está enfocado en la ciencias físicas [...] y el Centro Médico Ronald Reagan de UCLA.

Zh — 位于大学占地东半部的校园名称是什么?

整个校园被不正式地分为***南北两个校园***，这两个校园都位于大学占地的东半部。北校园是原校园的中心，建筑以义大利文艺复兴时代建筑闻名，其中的包威尔图书馆（Powell Library）成为好莱坞电影的最佳拍摄场景。[...] 这个广场曾在许多电影中出现。

Hi — विश्वविद्यालय जहाँ स्थित है, उसके पूर्वी दिशा में बने परिसरों को क्या नाम दिया गया है?

जब 1919 में यूसीएलए ने अपना नया परिसर खोला, तब इसमें चार इमारतें थी। [...] परिसर अनौपचारिक रूप से ***उत्तरी परिसर और दक्षिणी परिसर*** में विभाजित है, जो दोनों विश्वविद्यालय की जमीन के पूर्वी हिस्से में स्थित हैं। [...] दक्षिणी परिसर में भौतिक विज्ञानं, जीव विज्ञान, इंजीनियरिंग, मनोविज्ञान, गणितीय विज्ञान, सभी स्वास्थ्य से संबंधित क्षेत्र और यूएलसीए मेडिकल सेंटर स्थित है।

(b) MLQA example parallel for En-Es-Zh-Hi

Figure 4.3: MLQA examples. Answers shown as highlighted spans in contexts. Contexts shortened for clarity with "[...]".

fold	en	de	es	ar	zh	vi	hi
dev	1148	512	500	517	504	511	507
test	11590	4517	5253	5335	5137	5495	4918

(a) No. instances per language

	de	es	ar	zh	vi	hi
de	5029					
es	1972	5753				
ar	1856	2139	5852			
zh	1811	2108	2100	5641		
vi	1857	2207	2210	2127	6006	
hi	1593	1910	2017	2124	2124	5425

(b) No. parallel instances per language pair.

Table 4.2: Number of instances in MLQA. (All instances parallel with English)

Table 4.4 shows statistics about the lengths of contexts, questions and answers in MLQA. Vietnamese has the longest contexts on average and German are shortest, but all languages have a substantial tail of long contexts. Other than Chinese, answers are on average 3 to 4 tokens.

To investigate the distribution of topics in MLQA, a random sample of 500 articles were manually analysed. Articles cover a broad range of topics across different cultures, world regions and disciplines. 23% are about people, 19% on physical places, 13% on cultural topics, 12% on science/engineering, 9% on organisations, 6% on events and 18% on other topics.

	en	de	es	ar	zh	vi	hi
# Articles	5530	2806	2762	2627	2673	2682	2255
# Contexts	10894	4509	5215	5085	4989	5246	4524
# Instances	12738	5029	5753	5852	5641	6006	5425

Table 4.3: Number of Wikipedia articles with a context in MLQA.

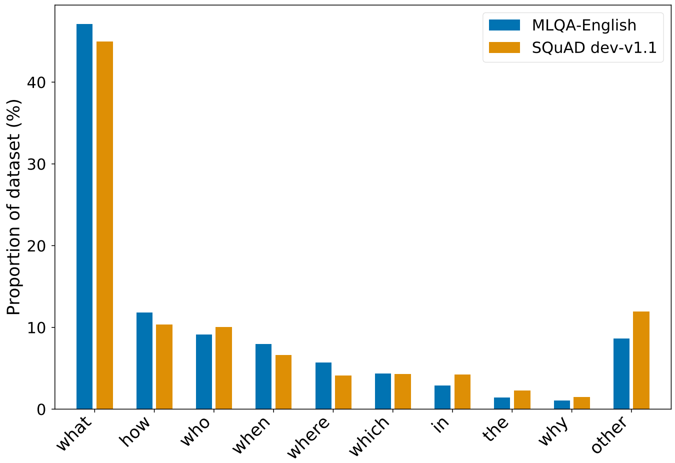


Figure 4.4: Question type distribution (by wh*-word) in MLQA-en and SQuAD V1.1. The distributions are strikingly similar

4.3 Cross-lingual RC Experiments

We introduce two tasks to assess cross-lingual RC performance with MLQA. The first, *cross-lingual transfer* (XLT), requires training a model with (c_x, q_x, a_x) training data in language x, in our case English. Development data in language x is used for tuning. At test time, the model must extract answer a_y in language y given context c_y and question q_y. The second task, *generalised cross-lingual transfer* (G-XLT), is trained in the same way, but at test time the model must extract c_z from c_z in language z given q_y in language y. This evaluation setup is possible because MLQA is highly parallel, allowing us to swap q_z for q_y for parallel instances without changing the question's meaning.

As MLQA only has development and test data, we adopt SQuAD v1 as training data. We use MLQA-en as development data, and focus on zero-shot evaluation, where no training or development data is available in target languages. Models were trained with the SQuAD-v1 training method from Devlin et al. (2019) and

	en	**de**	**es**	**ar**	**zh***	**vi**	**hi**
Context	157.5	102.2	103.4	116.8	222.9	195.1	141.5
Question	8.4	7.7	8.6	7.6	14.3	10.6	9.3
Answer	3.1	3.2	4.1	3.4	8.2	4.5	3.6

Table 4.4: Mean Sequence lengths (tokens) in MLQA. *calculated with mixed segmentation (section 4.3.1)

implemented in Pytext (Aly et al., 2018).

We experiment with 3 different modelling techniques to assess current cross-lingual RC capabilities, detailed below:

Translate-Train We translate instances from the SQuAD training set into the target language using machine-translation.[4] Before translating, we enclose answers in quotes, as in Lee et al. (2018b). This makes it easy to extract answers from translated contexts, and encourages the translation model to map answers into single spans. We discard instances where this fails (∼5%). This dataset is then used to train an RC model in the target language.

Translate-Test In this setting, the context and question in the target language is translated into English at test time, and we then use our best English RC model to produce an answer in the translated paragraph. We then need to translate this English answer back into the target language. For all languages other than Hindi,[5] we use attention scores, A_{ij}, from the translation model to map the answer back to the original language. Rather than aligning spans by attention argmax, as by Asai et al. (2018), we find it beneficial to use the span in the original context which

[4] We use Facebook's production translation models from July 2019.

[5] Alignments were unavailable for Hindi-English due to production model limitations. Instead we translate English answers using another round of translation. Back-translated answers may not map back to spans in the original context, so this Translate-Test performs poorly.

maximises F1 score with the English span:

$$
\begin{aligned}
\mathrm{RC} &= \textstyle\sum_{i \in a_{en}, j \in a_o} A_{ij} \big/ \sum_{i \in a_{en}} A_{i*} \\
\mathrm{PR} &= \textstyle\sum_{i \in a_{en}, j \in a_o} A_{ij} \big/ \sum_{j \in a_o} A_{*j} \\
\mathrm{F1} &= 2 * \mathrm{RC} * \mathrm{PR} \big/ \mathrm{RC} + \mathrm{PR} \\
\text{answer} &= \operatorname*{arg\,max}_{a_o} \ \mathrm{F1}(a_o)
\end{aligned}
\tag{4.1}
$$

where a_{en} and a_o are the English and target language answer spans respectively, $A_{i*} = \sum_j A_{ij}$ and $A_{*j} = \sum_i A_{*j}$.

Cross-lingual Representation Models Here, we experiment with language-agnostic pretrained language models. These models are trained with background corpora from many languages, and have been shown to learn language-agnostic behaviour, i.e. these models can be finetuned in one language for a classification task such as NLI, and still perform this task in a different language (Conneau and Lample, 2019). We produce zero-shot transfer results from M-BERT (cased, 104 languages) (Devlin et al., 2019) and XLM (MLM + TLM, 15 languages) (Conneau and Lample, 2019). Models are trained with the SQuAD training set and evaluated directly on the MLQA test set in the target language. Model selection is also constrained to be strictly zero-shot, using only English development. As a result, we end up with a single model that we test for all 7 languages.

4.3.1 Evaluation Metrics for Cross-lingual RC

As detailed in Section 2.6 in chapter 2, most RC tasks use EM and mean F1 metrics. We introduce the following modifications for fairer multilingual evaluation: Instead of stripping only ASCII punctuation when normalising answers, we strip all unicode characters with a punctuation *General_Category*.[6] In addition, we strip standalone articles for languages which have them (English, Spanish, German and Vietnamese). We use whitespace tokenisation for all MLQA languages other than Chinese, where we use the mixed segmentation method from Cui et al. (2019b).

[6] http://www.unicode.org/reports/tr44/tr44-4.html#General_Category_Values

F1 / EM	en	es	de	ar	hi	vi	zh
BERT-L	**80.2 / 67.4**	-	-	-	-	-	-
M-BERT	77.7 / 65.2	64.3 / 46.6	57.9 / 44.3	45.7 / 29.8	43.8 / 29.7	57.1 / 38.6	57.5 / 37.3
XLM	74.9 / 62.4	**68.0 / 49.8**	**62.2 / 47.6**	**54.8 / 36.3**	48.8 / 27.3	61.4 / 41.8	61.1 / **39.6**
Trans-test BERT-L		65.4 / 44.0	57.9 / 41.8	33.6 / 20.4	23.8 / 18.9*	58.2 / 33.2	44.2 / 20.3
Trans-train M-BERT		53.9 / 37.4	62.0 / 47.5	51.8 / 33.2	**55.0 / 40.0**	**62.0 / 43.1**	**61.4** / 39.5
Trans-train XLM		65.2 / 47.8	61.4 / 46.7	54.0 / 34.4	50.7 / 33.4	59.3 / 39.4	59.8 / 37.9

Table 4.5: F1 and EM scores on the MLQA test set for the cross-lingual transfer task (XLT)

4.4 Results

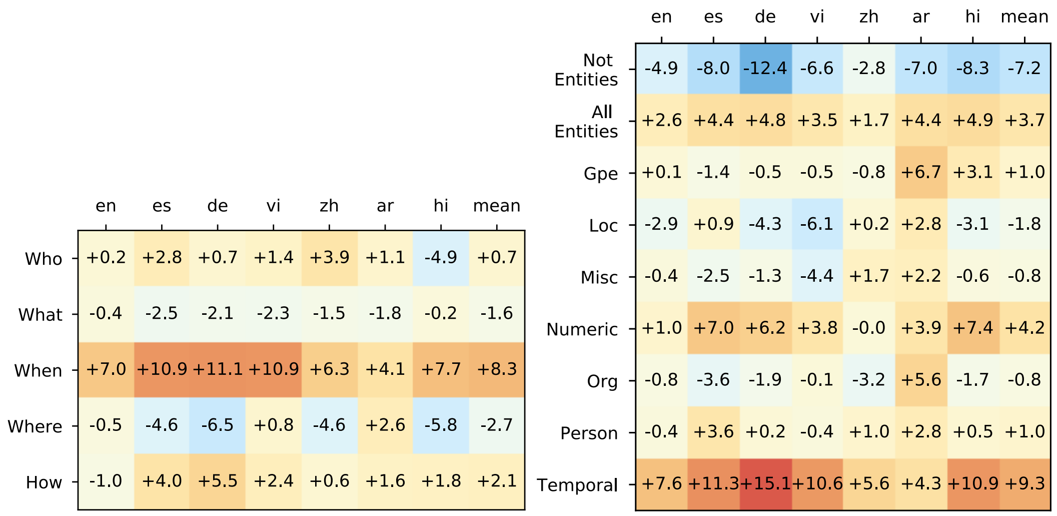
4.4.1 XLT Results

Table 4.5 shows the results on the XLT task. XLM performs best overall, transferring best in Spanish, German and Arabic, and competitively with translate-train+M-BERT for Vietnamese and Chinese. XLM is however, weaker in English. Even for XLM, there is a 39.8% drop in mean EM score (20.9% F1) over the English BERT-large baseline, showing significant room for improvement. All models generally struggle on Arabic and Hindi.

A manual analysis of cases where XLM failed to exactly match the gold answer was carried out for all languages. 39% of these errors were completely wrong answers, 5% were annotation errors and 7% were acceptable answers with no overlap with the gold answer. The remaining 49% come from answers that partially overlap with the gold span. The variation of errors across languages was small.

To examine how performance varies across languages for different types of question, we stratify MLQA with three criteria — By English wh*-word, by answer Named-Entity type and by English Question Difficulty

By wh*-word: First, we split by the English wh*-word in the question. This resulting change in F1 score compared to the overall F1 score is shown in Figure 4.5a, and discussed briefly in the main text. The English wh* word provides a clue as to the type of answer the questioner is expecting, and thus acts as a way of classifying RC instances into types. We see that the model finds “when” questions consistently easier than average across the languages, but the pattern is less consistent for other



(a) F1 stratified by English wh* word **(b)** F1 stratified by named entity types in answers

Figure 4.5: F1 Scores stratified by different criteria relative to overal F1 Scores for XLM

question types. “Where” questions seem challenging for Spanish, German, Chinese and Hindi, but this is not true for Arabic or Vietnamese.

By Named-Entity type We create subsets of MLQA by detecting which English named entities are contained in answer spans using an off-the-shelf named entity recogniser (Honnibal et al., 2019). The F1 scores relative to overall F1 score are shown for various Named Entity types in Figure 4.5b. There are some clear trends: Answer spans that contain named entities are easier to answer than those that do not (the first two rows) for all the languages, but the difference is most pronounced for German. Secondly,“Temporal” answer types (DATE and TIME entity labels) are consistently easier than average for all languages, consistent with the high scores for “when” questions which we saw above. Again, this result is most pronounced in German, but is also very strong for Spanish, Hindi, and Vietnamese. Arabic also performs well for ORG, GPE and LOC answer types, unlike most of the other languages. Numeric questions (CARDINAL, ORDINAL, PERCENT, QUANTITY and MONEY types) also seem relatively easy for the model in most languages.

By English Question Difficulty Here, we split MLQA into two subsets, according to whether the XLM model got the question completely wrong in English (no word overlap with the correct answer). We then evaluated the mean F1 score for each language on the two subsets, with the results shown in Figure 4.6. We see that

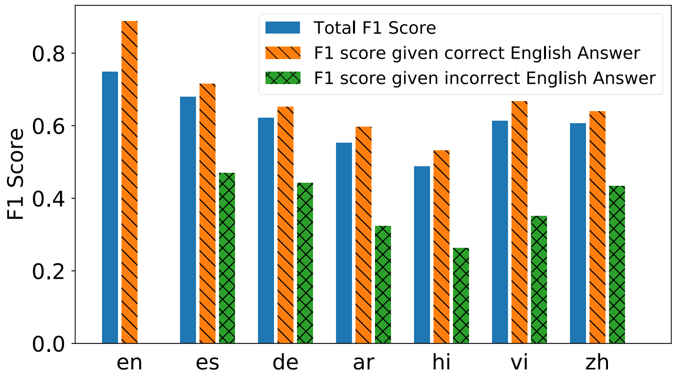


Figure 4.6: XLM F1 score stratified by English difficulty

questions that are “easy” in English also seem to be easier in the target languages, but the drop in performance for the “hard” subset is not as dramatic as one might expect. This suggests that not all questions that are hard in English in MLQA are hard in the target languages. This could be due to the grammar and morphology of different languages leading to questions being easier or more difficult to answer. Another factor is that context documents can differ in target languages for questions the model struggled to answer correctly in English, for example by being shorter, which may make them easier. Manual inspection suggests that whilst context documents are often shorter for when the model is correct in the target language, this effect is not sufficient to explain the difference in performance.

4.4.2 G-XLT Results

Tables 4.6a and 4.6a shows results for XLM and M-BERT on the G-XLT task. XLM outperforms M-BERT for most language pairs, with a mean G-XLT performance of 53.4 F1 compared to 47.2 F1 (mean of off-diagonal elements of Tables 4.6a and 4.6b). For questions in a given language, XLM performs best when the context language matches the question, except for Hindi and Arabic. For contexts in a given language, English questions tend to perform best, apart from for Chinese and Vietnamese. M-BERT exhibits more of a preference for English than XLM for G-XLT, and exhibits a bigger performance drop going from XLT to G-XLT (10.5 mean drop in F1 compared to 8.2).

c/q	en	es	de	ar	hi	vi	zh
en	74.9	65.0	58.5	50.8	43.6	55.7	53.9
es	69.5	68.0	61.7	54.0	49.5	58.1	56.5
de	70.6	67.7	62.2	57.4	49.9	60.1	57.3
ar	60.0	57.8	54.9	54.8	42.4	50.5	43.5
hi	59.6	56.3	50.5	44.4	48.8	48.9	40.2
vi	60.2	59.6	53.2	48.7	40.5	61.4	48.5
zh	52.9	55.8	50.0	40.9	35.4	46.5	61.1

(a) XLM

c/q	en	es	de	ar	hi	vi	zh
en	77.7	64.4	62.7	45.7	40.1	52.2	54.2
es	67.4	64.3	58.5	44.1	38.1	48.2	51.1
de	62.8	57.4	57.9	38.8	35.5	44.7	46.3
ar	51.2	45.3	46.4	45.6	32.1	37.3	40.0
hi	51.8	43.2	46.2	36.9	43.8	38.4	40.5
vi	61.4	52.1	51.4	34.4	35.1	57.1	47.1
zh	58.0	49.1	49.6	40.5	36.0	44.6	57.5

(b) M-BERT

Table 4.6: F1 Scores on the G-XLT Task. Columns show question language, rows show context language.

Model	**SQuAD**	**SQuAD***	**MLQA**-en
BERT-Large	91.0 / 80.8	84.8 / 72.9	80.2 / 67.4
M-BERT	88.5 / 81.2	83.0 / 71.1	77.7 / 65.1
XLM	87.6 / 80.5	82.1 / 69.7	74.9 / 62.4

Table 4.7: English performance comparisons (F1 / EM) to SQuAD using our models. * uses a single answer annotation.

4.4.3 English Results on SQuAD v1 vs MLQA

The MLQA-en results in Table 4.5 are lower than reported results on SQuAD v1.1 in the literature for equivalent models. However, once SQuAD scores are adjusted to reflect only having one answer annotation (picked using the same method used to pick MLQA answers), the discrepancy drops to 5.8% on average (see Table 4.7). MLQA-en contexts are on average 28% longer than SQuAD's, and MLQA covers a much wider set of articles than SQuAD. Minor differences in preprocessing and answer lengths may also contribute (MLQA-en answers are slightly longer, 3.1 tokens vs 2.9 on average). Question type distributions are very similar in both datasets, as shown in Figure 4.4.

4.5 Related Work

Here we shall highlight some related work of specific interest to this chapter, which hasn't been discussed in previous chapters.

Monolingual RC Data Large, high-quality datasets in languages other than English are relatively rare. There are several Chinese datasets, e.g. DUReader (He et al., 2018), CMRC (Cui et al., 2019b) and DRCD (Shao et al., 2018). Recently, there have been efforts in a wider array of languages, such as Korean (Lim et al., 2019), French (d'Hoffschmidt et al., 2020), Arabic (Mozannar et al., 2019), German (Möller et al., 2021) and Hebrew (Keren and Levy, 2021).

Cross-lingual QA Modelling Cross-lingual information retrieval has been studied for decades, driven by the CLEF workshops in the early-to-mid 2000s (Peters, 2001). Cross-lingual QA for RDF data for a number of years, such as the QALD-3 and 5 tracks (Cimiano et al., 2013; Unger et al., 2015), with more recent work from Zimina et al. (2018). The modern RC format has a comparatively short history. Lee et al. (2018b) explore an approach to use English RC data from SQuAD to improve RC performance in Korean using an in-language seed dataset. Kumar et al. (2019) study question generation by leveraging English questions to generate better Hindi questions, and Lee and Lee (2019) and Cui et al. (2019a) develop modelling approaches to improve performance on Chinese RC tasks using English resources. Lee et al. (2019b) and Hsu et al. (2019) explore modelling for zero-shot transfer and Singh et al. (2019) explore regularising RC models with cross-lingual data.

Cross-lingual RC Data Gupta et al. (2018) release a parallel RC dataset in English and Hindi, Hardalov et al. (2019) investigate RC transfer from English to Bulgarian, Liu et al. (2019b) release a cloze RC dataset in Chinese and English, and Jing et al. (2019) released BiPar, built using parallel paragraphs from novels in English and Chinese. These datasets have a similar spirit to MLQA, but are limited to two languages. Asai et al. (2018) investigate RC on a manually translated set of 327 SQuAD instances in Japanese and French, and develop a phrase-alignment modelling technique, showing improvements over back-translation. Like us, they build multi-way parallel RC data, but MLQA has many more instances, covers more languages and does not require manual document translation. Liu et al. (2019a) explore a kind of cross-lingual open-domain cloze QA with a dataset built from Wikipedia "Did you know?" questions, covering nine languages. Unlike MLQA, it is distantly supervised, the dataset size varies by language, instances are not parallel, and answer distributions vary by language, making quantitative comparisons across languages challenging. Finally, in contemporary work, Artetxe et al. (2020) release

XQuAD, a dataset of 1190 SQuAD instances from 240 paragraphs manually translated into 10 languages. As shown in Table 4.3, MLQA covers 7 languages, but contains more data per language – over 5k RC pairs from ∼5k paragraphs per language. MLQA also uses real Wikipedia contexts rather than manual translation.

Aggregated Cross-lingual Benchmarks Recently, following the widespread adoption of projects such as GLUE (Wang et al., 2018a), there have been efforts to compile a suite of high quality multilingual tasks as a unified benchmark system. Two such projects, XGLUE (Liang et al., 2020) and XTREME (Hu et al., 2020) incorporate MLQA as part of their aggregated benchmark.

4.6 Reflection

It is worth discussing the quality of context paragraphs in MLQA. Our parallel sentence mining approach can source independently written documents in different languages, but, in practice, articles are often translated from English to the target languages by volunteers. Thus our method sometimes acts as an efficient mechanism of sourcing existing human translations, rather than sourcing independently written content on the same topic. The use of machine translation is strongly discouraged by the Wikipedia community,[7] but from examining edit histories of articles in MLQA, machine translation is occasionally used as an article seed, before being edited and added to by human authors.

Our annotation method restricts answers to come from specified sentences. Despite being provided with several sentences of context, some annotators may be tempted to only read the parallel sentence and write questions which only require a single sentence of context to answer. However, single-sentence context questions are a known issue in SQuAD annotation in general (Sugawara et al., 2018) suggesting our method would not result in less challenging questions, supported by scores on MLQA-en being similar to SQuAD (section 4.4.3).

MLQA is partitioned into development and test splits. As MLQA is parallel, this

[7] https://en.wikipedia.org/w/index.php?title=Wikipedia:Translation&oldid=888723630#Avoid_machine_translations

means there is development data for every language. Since MLQA's test set is freely available, this was done to reduce the risk of test set over-fitting that may occur in future, and to establish standard splits. However, in our experiments, we only make use of the English development data and study strict zero-shot settings. Other evaluation setups could be envisioned, e.g. by exploiting the target language development sets for hyper-parameter optimisation or fine-tuning, which could be fruitful for higher transfer performance, but we leave such "few-shot" experiments as future work. Other potential areas to explore involve training datasets other than English, such as CMRC (Cui et al., 2019b), or applying using unsupervised techniques such as those in chapter 3 to assist transfer.

Finally, as we have discussed in chapters 2 and 3, a large body of work suggests RC models can be over-reliant on word-matching between question and context. G-XLT represents an interesting test-bed, as simple symbolic matching is less straightforward when questions and contexts use different languages. However, the drop from XLT is relatively small (8.2 F1), suggesting word-matching in cross-lingual models is more nuanced and sub-symbolic than it may initially appear.

4.7 Conclusion

In this chapter, we have introduced MLQA, a highly parallel multilingual RC benchmark on seven diverse languages. We evaluated several different modelling approaches on two cross-lingual understanding tasks on MLQA. We found that pretrained cross-lingual models, specifically XLM, outperformed translation-based approaches, and represent a promising way forward for multilingual QA systems. Not only does this approach lead to the highest accuracy, it also avoids the need for slow and potentially error-prone machine translation, and only a single model checkpoint is required at test time to support many languages.

Since the release of MLQA, some new resources and datasets have become available. The most notable is TyDiQA (Clark et al., 2020), which addresses some of the limitations of MLQA, such as not having unanswerable questions, inheriting known weaknesses from the SQuAD annotation protocol and using (modest amounts) of manual translation. This dataset does have a couple of different drawbacks relative to our work, such as not having parallel instances, and resultant issues with cross-language comparisons. These two datasets thus complement each other well.

Cross-lingual technology has also improved since the experiments in this chapter were carried out. Cross-lingual pretrained models continue to be the dominant method of choice for multi-lingual QA. Models such as XLMR (Conneau et al., 2020) improve the average XLT F1 score from 61.6 to 71.6, and the largest mT5 model (Xue et al., 2021) increase it even further to 76.0. Remarkably, mT5's average F1 score across all the languages is higher than XLM's English F1 score. Whilst there is still a gap between training language and testing languages, it continues to shrink quickly as cross-lingual models improve.

In Part I, we have explored how to tackle RC tasks in zero-shot settings – first, in chapter 3 with no training RC data at all, and then, in chapter 4 in a more practically relevant setting where we have no in-language training RC data, but are otherwise unrestricted. However, the pure RC task, where we are provided with a short paragraph of text guaranteed to contain the answer to a question, is a relatively rare circumstance in practical settings. Moreover, this task does not place significant demands on our ability to leverage knowledge from larger knowledge sources. Thus, in the next chapter (and the remainder of the thesis) we shall shift to considering open-domain QA, dropping the assumption that an oracle context document will be provided for us. That being said, our findings from chapters 3 and 4 will remain relevant, especially for retrieve-and-read ODQA architectures, which utilise RC components as part of their design.

Part II

Retrieval-augmented Pretrained Models

"The only thing that you absolutely have to know, is the location of the library."

Albert Einstein

Chapter 5

How Context Affects Language Models' Factual Predictions

In section 2.3.2.4 in chapter 2, we introduced the concept of *parametric knowledge*. This is the phenomenon that large, pretrained language models capture relational knowledge expressed in their pretraining corpus. Part II of this thesis will focus on parametric knowledge – how it compares to, and can be complemented by, more traditional *non-parametric* knowledge access mechanisms, such as retrieval. For the remainder of this thesis, we shall shift away from the RC problems we studied in Part I, and use open-domain QA tasks (section 2.4.2) as a test-bed for investigating models which can better leverage knowledge. However, RC mechanisms will still remain relevant, as a key component of retrieve-and-read ODQA models (section 2.4.2.1), which feature prominently throughout this chapter, and the next.

Relational parametric knowledge can be "retrieved" in a sense, by presenting models with cloze questions (Section 2.1.3), and treating the filled-in blank as an answer. The LAMA probe (Petroni et al., 2019) is a collection of relational-knowledge open-domain cloze questions, which we shall use to measure the extent of factual relational knowledge in models. Petroni et al. (2019) have used LAMA to demonstrate that models like BERT store substantial relational knowledge. However, considering the millions of documents and facts in a knowledge source like Wikipedia, it is unlikely that a model with a finite and practical number of parameters will be able to reliably store and retrieve factual knowledge with sufficient precision to be useful. Even if this is possible, such a model may have to be extremely large, and editing and updating the knowledge of the model will be challenging. One potential way forward is to augment the model with retrieved passages from an IR system, turning it into a kind of retrieve-and-read model.

In this chapter, we shall study the purely unsupervised case of augmenting a language model with retrieved contexts at test time, to assess whether (unsupervised) parametric-knowledge systems can benefit from the addition of (unsupervised) non-parametric knowledge. We demonstrate that augmenting pretrained language models with retrieved contexts dramatically improves unsupervised cloze QA on the LAMA probe, reaching performance on par with DRQA, a popular *supervised* ODQA baseline. In addition to being unsupervised, using a pretrained language model like BERT instead of a trained RC model has several other potential advantages. Since BERT is not a span-extractive model, it is able to utilise contexts that contain relevant information but do not contain the answer span directly (although we do not find strong quantitative evidence or this in this chapter) . More importantly, we find that, via the next-sentence prediction objective, BERT is able to ignore noisy or irrelevant contexts.

In summary, we shall present the following core findings in this chapter:

- Augmenting queries with relevant contexts dramatically improves BERT and RoBERTa performance on LAMA (Petroni et al., 2019), demonstrating the unsupervised RC capabilities of pretrained language models

- Fetching contexts using an off-the-shelf information retriever is sufficient for BERT to match the performance of an early supervised ODQA baseline

- BERT's next-sentence prediction pretraining strategy is a highly effective unsupervised mechanism in dealing with noisy and irrelevant contexts

Bigger Picture This chapter shifts focus from RC, which was the main task studied in the last two chapters, towards open-domain QA. This chapter does however, build thematically on the previous chapters by continuing the focus on the low-data regimes, and we will take a deep look at how to get models to produce more factually accurate predictions without training data. This change in task is accompanied by a requirement that models store and access much more knowledge than is required in RC, where the knowledge required to answer questions is provided to the model in the form of a context document. Thus, how knowledge is stored, represented and retrieved becomes of great importance. This chapter introduces the concept of comparing parametric and non-parametric knowledge, which we will place a great emphasis on, and use as a lens for understanding QA, in all of the remaining chapters. Chapter 6 will directly build on the retrieval-augmented models we propose and study in this chapter, and chapters 7 and 8 will examine how different knowledge modelling architecture choices can lead to different levels of generalization and QA behaviour. Additional commentary on the connections between this chapter and the wider body of work in the thesis can be found in the conclusion of this chapter (sec. 5.5) and the thesis conclusion, chapter 9.

The material in this chapter first appeared in:

> Fabio Petroni, **Patrick Lewis**, Aleksandra Piktus, Tim Rockäschel, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2020. How Context Affects Language Models' Factual Predictions. In *Proceedings of the 2nd Annual Automated Knowledge Base Construction Conference (AKBC)*

> *Individual Contributions: The initial observation of contexts improving results on LAMA was made by the lead author. The experimental design was a close collaboration between the lead author and the thesis author, with additional input from co-authors. The oracle and retrieval experiments were performed by co-authors. The thesis author proposed and implemented the adversarial context and NSP experiments. The original paper was written as a collaboration between co-authors, with thesis author writing substantial portions.*

Corpus	Relation	Statistics	
		# Facts	# Relations
Google-RE	birth-place	2937	1
	birth-date	1825	1
	death-place	765	1
	Total	5527	3
T-REx	1-1	937	2
	N-1	20006	23
	N-*M*	13096	16
	Total	34039	41
SQuAD	Total	305	-

Table 5.1: Statistics for the LAMA data.

5.1 Methodology

Given a cloze question q with an answer a, we assess how the predictions from a language model change when we augment the input with contexts c. In this section, we describe the datasets we use to source (q,a) pairs, as well as various methods of selecting context documents c.

5.1.1 Datasets

We use the LAMA probe in our experiments in this chapter (Petroni et al., 2019). LAMA is a collection of cloze questions about real world relational facts, which all have a single token answer. Each question is accompanied by snippets of text from Wikipedia that are likely to express the corresponding fact.

Although there are a number of cloze QA datasets we could choose (some are listed later in Section 5.3) we use LAMA because: (1) the nature of the LAMA data is aligned with our relational knowledge focus (*i.e.*, given a subject and a relation predict the object) and (2) each data point is aligned by construction with relevant context passages, which will provide a useful “oracle” context evaluation setting.

LAMA is comprised of data from several sources. For the experiments in this chapter, we use the Google-RE, T-REx and SQuAD subsets, which are briefly described below. Further details on LAMA can be found in Petroni et al. (2019).

Google-RE The Google-RE[1] dataset contains ∼60K facts manually extracted from Wikipedia. It covers five relations but LAMA only considers three of them, namely "PLACE OF BIRTH", "DATE OF BIRTH" and "PLACE OF DEATH". The other two are excluded since they contain mainly multi-tokens objects which are not supported in LAMA evaluation. A manually defined cloze question template is used for each relation, e.g., "[S] was born in the city of [O]" for "PLACE OF BIRTH". Each fact in the Google-RE dataset is, by design, manually aligned to a short passage in Wikipedia which supports it.

T-REx The T-REx dataset is a subset of Wikidata triples. It is derived from the T-REx dataset (Elsahar et al., 2018) and is much larger than Google-RE with a broader set of relations. LAMA consider 41 Wikidata relations and subsamples to at most 1000 facts per relation. As with Google-RE, a manual template for each relation is used to map triples to cloze questions. In contrast to Google-RE , T-REx facts were automatically aligned to Wikipedia, and there is some potential for noise, although Elsahar et al. (2018) report the error rate to be a very low 2.2%

SQuAD LAMA also contains a carefully selected subset of 305 context-insensitive questions from the SQuAD development set with single token answers. Each question is manually mapped to a cloze-style question, e.g. by mapping "Who developed the theory of relativity?" as "The theory of relativity was developed by [MASK]". Each question-answer pair is accompanied by the passage in Wikipedia it was initially annotated on. As noted in section 2.5.2.2, SQuAD is not appropriate for ODQA tasks in general. However, the questions selected for LAMA were specifically chosen for context-insensitivity, and manually verified to be appropriate.

Detailed statistics for the LAMA data considered in this chapter are reported in Table 5.1. For the RoBERTa results, we trim the LAMA dataset (by about 15%) such that all answers are single tokens in the model's vocabulary, so BERT and RoBERTa numbers in this chapter should not be directly compared as they consider slightly different subsets of the data.

[1] https://code.google.com/archive/p/relation-extraction-corpus

5.1.2 Baselines

We consider DRQA (Chen et al., 2017), a popular baseline for ODQA. DrQA is a retrieve-and-read model, (see section 2.4.2.1) consisting of a TF-IDF retriever over Wikipedia text and an RC reader trained on SQuAD v1. In order to apply DRQA to the LAMA probe, we take inspiration from Levy et al. (2017) and map each cloze template to a natural question template (*e.g.*, "X was born in [MASK]" to "Where was X born?"). We constrain the span predictions of DRQA to single-token answers, for fairer comparison to the mask-filling language models.

5.1.3 Language Models

We consider BERT-LARGE-CASED (Devlin et al., 2019) and ROBERTA-LARGE (Liu et al., 2019c) in our experiments. Both BERT and ROBERTA have been trained on corpora that include Wikipedia. While BERT uses two pretraining strategies, Mask Language Modelling (MLM) and Next Sentence Prediction (NSP), ROBERTA considers only the MLM task. We compute a probability distribution over the unified vocabulary of Petroni et al. (2019) for the masked token for each cloze question, take the argmax token as the prediction, and report EM scores.

5.1.4 Contexts

We augment the cloze questions with different types of contextual information. We explicitly distinguish cloze question q and context c in the input according to the model. For BERT, we use different segment embeddings, index 0 for q and 1 for c, and insert the separator token (*i.e.*, [SEP]) in between. For ROBERTA, which is not equipped with segment embeddings, we use the end of sentence (EOS) token to separate q and c.

Oracle Contexts We provide an oracle-based (ORA) context in order to assess the capability of LMs to exploit context that we *know* is relevant to the entity in the question. Concretely, we use the gold Wikipedia passage which accompanies each example in the LAMA probe, truncated to at most five sentences. This context often contains the true answer and always contains helpful related information.

LAMA	Relation			*automatically-sourced context*		
		B	B-ADV	DrQA	B-RET	B-ORA
Google-RE	birth-place	16.1	14.5	**48.6**	43.5	*70.6*
	birth-date	1.4	1.4	42.9	**43.1**	*98.1*
	death-place	14.0	12.6	**38.4**	35.8	*65.1*
	Total	10.5	9.5	**43.3**	40.8	*78.0*
T-REx	1-1	74.5	74.5	55.2	**81.2**	*91.1*
	N-1	34.2	33.8	30.4	**47.5**	*67.3*
	N-*M*	24.3	23.6	15.4	**32.0**	*52.4*
	Total	32.3	31.8	25.8	**43.1**	*62.6*
SQuAD		17.4	17.4	**37.5**	34.3	*61.7*
weighted average		30.5	30.0	27.2	**42.8**	*63.6*

Table 5.2: EM scores for the DrQA baseline, BERT-large on context-free questions (B), on adversarial (B-ADV), retrieved (B-RET) and oracle (B-ORA) context-augmented questions. The unsupervised B-RET is competitive with the supervised DrQA and much stronger than the context-free baseline. We weight the average per number of relations. Pairwise sign tests show statistically significant differences ($p < 1e\text{-}5$) between: B-RET and all other results; B-ORA and all other results.

Sourcing Relevant Contexts For ODQA, known-to-be-relevant context documents are not available and must be automatically sourced. Here, we use DrQA's TF-IDF retriever, which indexes Wikipedia. We could aggregate or marginalise results over many retrieved passages, but for simplicity, we just select the most relevant paragraph as context for the model. In chapter 6 we shall develop a model which aggregates over several retrieved passages.

Adversarial Contexts We provide an uninformative context in order to test the ability of the model to ignore irrelevant context that is not useful for answering the query. Here, for a given (q,a) pair, we generate an adversarial context c_{adv} by randomly sampling an oracle context from other questions q' which have the same relation type but a different answer a'. This results in a context document that refers to a different subject entity but contains a distracting and semantically plausible answer a'. Table 5.5 shows some examples of adversarial contexts.

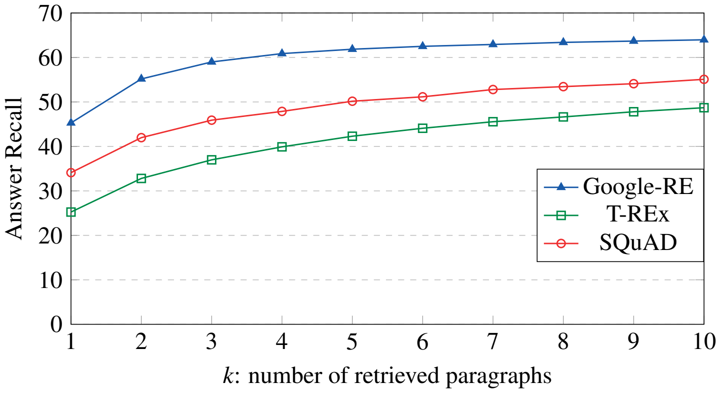


Figure 5.1: Percentage of times the answer appears in the top-*k* retrieved paragraphs.

EM score	*ans. in ctx.*	B-ADV	B-RET	B-ORA
better	*present*	0.9	14.0	32.6
	absent	2.4	3.2	1.4
	Total	3.3	17.2	34.0
worse	*present*	0.6	2.4	3.5
	absent	3.1	3.9	0.1
	Total	3.7	6.3	3.6
	# better relations	11	34	39

Table 5.3: Percentage of predictions which improve (*degrade*) when the context is provided for T-REx, grouped by the *presence* (*absence*) of the answer in the provided context. In general, B-RET and B-ORA score higher than the context-free model.

5.2 Results

The main results of our analysis are summarised in Table 5.2. It shows the LAMA probe EM score for the DRQA baseline and BERT-large augmented with different kinds of contextual information. Augmenting cloze questions with relevant context dramatically improves the performance of BERT: B-ORA obtains ×7.4 improvement on Google-RE, ×1.9 on T-REx and ×3.5 on SQuAD with respect context-free questions (B). This clearly demonstrates BERT's ability to successfully exploit provided context, effectively acting as an RC model. No fine-tuning is required to trigger such behaviour. Also, since there is no restriction for an answer to be a span from the context, this BERT setup is an *abstractive* RC model, capable of generating any answer in its vocabulary, unlike those seen so far in chapters 3-4.

5.2.1 Retrieval Augmentation

When we rely on TF-IDF-retrieved context (B-RET), BERT still performs much better than without augmentation. Overall, B-RET results are comparable with DRQA on Google-RE and SQuAD and much higher on T-REx. This is an intriguing, promising result given that B-RET, unlike DRQA, did not receive any supervision for this task. Pairwise sign tests across relations show that the improvements for B-RET and B-ORA are indeed statistically significant ($p<1e-5$). This result highlights that whilst BERT has impressive parametric knowledge capabilities, *combining* it with retrieved non-parametric knowledge has clear benefits.

Figure 5.1 shows the recall of the TF-IDF retriever, and shows that the answer is not always present in the top retrieved passage. This analysis indicates that aggregating predictions from many contexts would likely improve results further. Table 5.3 reports a detailed analysis of whether the answer is present in retrieved contexts and how that affects the model's predictions. We observe that most of the gain of B-RET comes from cases in which the context contained the answer. However, there are also cases where the context does not explicitly mention the answer but BERT is still able to utilise the related context to help select the correct answer (3.2%). Note that an extractive approach (such as DRQA) would have provided an incorrect answer (or no answer) for those cases. We shall take inspiration from this flexible answer generation in chapter 6, as well as incorporating a passage aggregation strategy. This being said, there are also cases where the model gets worse when the augmenting context document does not contain the answer. In fact, there are slightly more cases of the model moving from a correct answer to an incorrect answer than moving from a correct answer to an incorrect answer when the augmenting document doesn't contain the answer (3.9 vs 3.2% of cases). Thus, whilst this model has the ability to answer more questions correctly in theory, due to its open answer vocabulary, empirically there isn't strong evidence that this is happening effectively on the LAMA probe. We shall find evidence in chapter 6 of instance of this idea working well.

(% Next Sentence)	B-ADV	B-RET	B-ORA
Google-RE	10.4	88.9	98.4
T-REx	14.0	89.7	94.5
SQuAD	11.9	93.1	99.3

Table 5.4: % examples classified as 'next sentences' using BERT's NSP head. The low number of 'next sentence' classifications for B-ADV shows BERT recognises adversarial contexts as unrelated and thus limit their influence on predictions.

5.2.2 Adversarial Robustness

The B-ADV column in Table 5.2 shows the LAMA EM results for BERT for adversarial contexts. BERT is robust, dropping only 0.5 EM on average from the non-augmented baseline. However, as shown in Figure 5.2, this strong performance only occurs when the context and question are processed as two segments using BERT's separator tokens. Using only one segment (that is, simply concatenating the input query and the context) leads to a severe drop of 12.4 EM for BERT (a 40.7% relative drop in performance). We also observe a consistent improvement in performance from one segment to two for retrieved and oracle contexts.

One possible reason for this phenomenon resides in the Next Sentence Prediction (NSP) classifier of BERT, learned with self-supervision during pretraining by training the model to distinguish contiguous (*i.e.*, "next sentence" pairs) from randomly sampled blocks of text. We hypothesise that the MLM task might be influenced by the NSP's output. Thus, BERT might learn to not condition across segments for masked token prediction if the NSP score is low, thereby implicitly detecting irrelevant and noisy contexts. A result that seems in line with this hypothesis is that RoBERTa, which does not use NSP, is more vulnerable to adversarial contexts and the difference between one and two sentences (for RoBERTa separated by the EOS token) is much smaller.

To further investigate this hypothesis, we calculate the number of (c,q) pairs classified by BERT as "next sentence" pairs in LAMA for the different context strategies. These results are shown in Table 5.4. We see that for B-RET and B-ORA, NSP classifications are high, suggesting BERT finds the segments to be contigu-

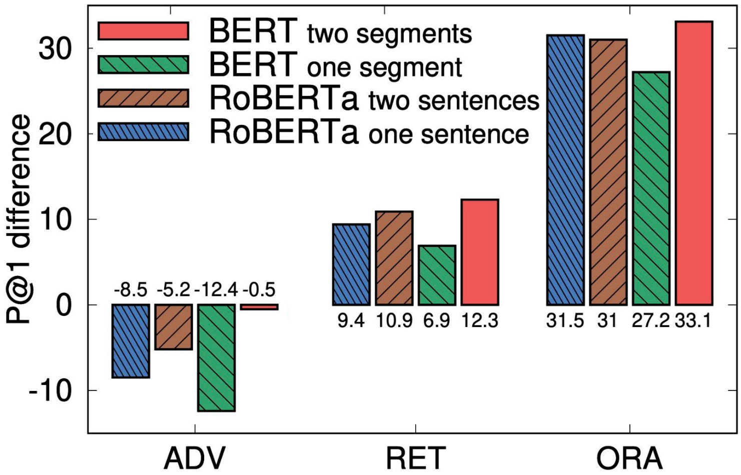
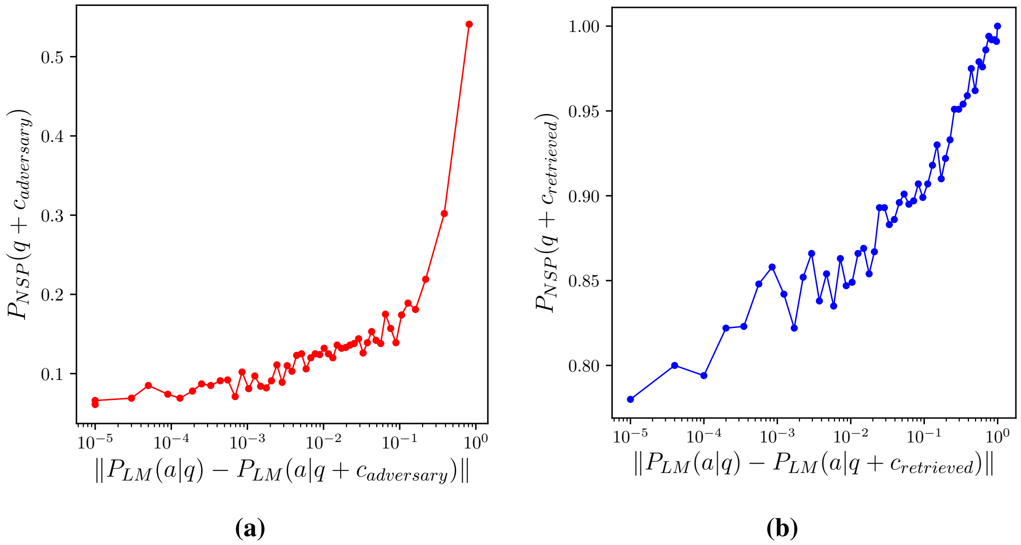


Figure 5.2: For each type of context, we report the change in EM score relative to no context, averaging results across relations. For each model we consider a concatenation of question and context as well as separating the two using separator tokens. Separation dramatically improves both model's ability to ignore poor context and improves BERT's performance in the presence of good context.



(a)

(b)

Figure 5.3: NSP Probabilities vs the change in LM probability using contexts from a) the adversary b) retrieval. The more relevance that BERT assigns the context, the greater the increase in likelihood of the correct answer. This is exactly what we would want if we had trained a relevance system ourselves, yet this instead emerges naturally from BERT's NSP pretraining.

ous, and hence useful to condition upon. However, for B-ADV, very few (c,q) pairs are classified as "next sentences", suggesting BERT may condition on them less. Additional evidence for our NSP adversarial robustness hypothesis is given in Figure 5.3. Here we compute the absolute difference in probability that BERT places on the correct answer upon including context $||P_{LM}(a|q) - P_{LM}(a|q+c)||_1$, and plot it against NSP probability $P_{NSP}(q,c)$. We see that for adversarial and retrieved contexts, increasing NSP probability is associated with greater change in true answer probability upon including context.[2]

Table 5.5 shows three examples for the generation of BERT-large for adversarial, retrieved and oracle context-augmented questions.

5.3 Related Work

Here we briefly review some related work on language model probing, which has not already covered in previous chapters. A variety of "probes" have been developed to analyse the syntactic structures in pretrained language models, such as syntax trees (Marvin and Linzen, 2018; Hewitt and Manning, 2019; Vig and Belinkov, 2019), negative polarity items (Warstadt and Bowman, 2019; Warstadt et al., 2019), semantic fragments (Richardson et al., 2020), function words (Kim et al., 2019), and many other linguistic phenomena (Tenney et al., 2019a,b). To measure the factual knowledge present in these pretrained language models, Petroni et al. (2019) propose the LAMA benchmark which tests the models with cloze questions constructed from knowledge triples, which we have used in this chapter. Jiang et al. (2020a) later extends LAMA by automatically discovering better prompts, Kassner and Schütze (2019) add negated statements, Poerner et al. (2019) filter out easy-to-guess queries, and Richardson and Sabharwal (2020); Talmor et al. (2020); Bisk et al. (2020) develop further probes for textual reasoning.

5.4 Reflection

Re-examining NSP The Next Sentence Prediction task has been extensively explored (Devlin et al., 2019; Liu et al., 2019c; Yang et al., 2019b; Lan et al., 2020) with the apparent consensus that it is not helpful for downstream fine-tuning accuracy. In contrast, our findings suggest that it is important for robust exploitation of retrieved context for unsupervised tasks. Basing design decisions on a limited

[2] Each context method has different NSP statistics, but the trend is consistent – higher NSP scores co-occur with greater changes in correct answer probability

	Query	Predictions
	[P101] Allan Sandage works in the field of ____ .	engineering [-3.1]
ADV	*q* [SEP] According to Gould, classical Darwinism encompasses three essential core commitments: Agency, the unit of selection, which for Charles Darwin was the organism, upon which natural ... [0.0]	psychology [-2.8] economics [-3.4] anthropology [-3.5]
RET	*q* [SEP] In 1922 John Charles Duncan published the first three variable stars ever detected in an external galaxy, variables 1, 2, and 3, in the Triangulum Galaxy (M33). These were followed up by Edwin ... [1.0]	**astronomy [-0.0]** physics [-5.5] observation [-7.3]
ORA	*q* [SEP] He currently works at the Institute of Astronomy in Cambridge; he was the Institute's first director.Educated at the University of Cambridge, in 1962 he published research with Olin Eggen ... [1.0]	**astronomy [-0.0]** physics [-4.0] galaxies [-5.5]
	[P279] Interleukin 6 is a subclass of ____ .	proteins [-0.2]
ADV	*q* [SEP]First built in 1893 by Chinese residents of Nagasaki with the support of the Qing Dynasty government, the shrine was designed to serve as a place of worship and learning for the Chinese ... [0.0]	proteins [-0.2] **protein [-3.1]** DNA [-3.7]
RET	*q* [SEP]In particular, the increase in levels of IL-6 (interleukin 6), a myokine, can reach up to one hundred times that of resting levels. Depending on volume, intensity, and other training factors ... [1.0]	insulin [-1.9] IL [-2.1] proteins [-2.4]
ORA	*q* [SEP]It is a cardiac hypertrophic factor of 21.5 kDa and a protein member of the IL-6 cytokine family. This protein heterodimerizes with interleukin 6 signal transducer to form the type II ... [1.0]	proteins [-0.7] **protein [-1.5]** insulin [-2.4]
	[P413] Giacomo Tedesco plays in ____ position .	center [-2.2]
ADV	*q* [SEP]On July 31, 2009 he was traded from the Tigers to the Seattle Mariners along with fellow pitcher Luke French for veteran pitcher Jarrod Washburn. On July 31, 2009 he was traded from ... [0.03]	center [-1.5] centre [-2.4] forward [-2.6]
RET	*q* [SEP]Giovanni Tedesco has two brothers who are also football players, Salvatore (formerly of Perugia and Lucchese) and Giacomo, who is playing for Reggina. [1.0]	**midfielder [-1.2]** forward [-1.8] midfield [-2.3]
ORA	*q* [SEP]Giacomo Tedesco (born Feb 1, 1976 in Palermo) is a former Italian football (soccer) midfielder and football manager. Giacomo Tedesco started his professional career... [1.0]	**midfielder [-0.7]** forward [-2.2] defender [-2.4]

Table 5.5: Examples of generation for BERT-large. We report the top three tokens predicted with the associated log probability (in square brackets) for adversarial (ADV), retrieved (RET) and oracle (ORA) context-augmented questions. NSP probability (in square brackets) reported at the end of each statement.

set of downstream tasks when designing general-purpose pretrained models may well us lead to less flexible models. As a community, we should continue to strive for greater diversity in our criteria and possible use-cases for assessing such models (Talmor et al., 2020).

Practical Takeaways Section 5.2 shows that BERT has a very different behaviour when inputs are processed with one or two segments. Practitioners should thus ensure that they thoroughly ablate segmentation options. The consistent improvement upon including retrieved context also suggests that it may be possible to get performance boosts in many other tasks by the incorporation of retrieved documents, even when such documents are not strictly required for the task. This will form the focus of chapter 6.

Limitations and Comparison with DRQA We demonstrate that BERT with retrieved context and no fine-tuning performs on par with DRQA on the LAMA probe, but it is worth discussing this comparison further. Firstly, it is encouraging that an unsupervised system performs just as well as a system that requires significant supervision such as DRQA. We further note that LMs are abstractive models, whereas DRQA is extractive, confined to returning answers that are spans of retrieved context. However, it is worth stating that LAMA only requires single token answers. Generating an arbitrarily long sequence of contiguous tokens from bidirectional LMs like BERT and ROBERTA is not trivial, but extractive QA models handle such cases by considering spans of text of varying lengths. Pretrained Seq2seq models alleviate this issue, and we shall employ them for multi-token answering on more standard ODQA benchmarks in chapter 6. Finally, whilst we have chosen DRQA as our baseline to compare to recent work, more sophisticated supervised ODQA models exist that outperform it on a variety of ODQA tasks. Moreover, the LAMA probe is not a traditional ODQA dataset, only covers quite specific relational knowledge, and does not feature a few key aspects that standard ODQA datasets capture, such as having natural questions. Thus, whilst the result here is promising, we cannot yet claim that unsupervised ODQA models are a competitive alternative to modern ODQA models in mainstream ODQA.

5.5 Conclusion

In this chapter, we demonstrated a simple technique to greatly improve factual unsupervised cloze QA by providing context documents as additional inputs. We used oracle documents to establish an upper bound to this improvement, and found that using an off-the-shelf retriever is sufficient to achieve performance on par with a supervised ODQA model. We also investigated how brittle models' factual predictions were to noisy and irrelevant context documents, and found that BERT, when featurised appropriately, is very robust. Finally, we provided evidence that this robustness stems from the Next Sentence Prediction pretraining task.

We interpret these results as evidence that retrieval-augmented language models are a powerful method of building knowledge-intensive systems, and that combining parametric and non-parametric knowledge is a promising direction for ODQA systems. That said, we also noted a number of limitations of our modelling set-up here. In particular, the retrieval-augmented systems we developed in this chapter are restricted to single-token answers, only consider a single passage of context, and rely on term-based retrieval to surface useful passages, which, as we saw in figure 5.1 is only partially successful. Moreover, we note that these results were obtained on the LAMA probe, using cloze questions, rather than on the more standard natural question-based ODQA datasets. In the next chapter, we shall address these issues, and shift from unsupervised probing of knowledge, towards investigating supervised settings.

Chapter 6

Retrieval-Augmented Generation for Knowledge-Intensive NLP

In the previous chapter, we explored how to combine parametric and non-parametric knowledge using the mechanism of retrieval-augmentation, demonstrating its efficacy on a knowledge-probing task. Such hybrids also have a range of additional benefits: their knowledge can be directly revised and expanded, and accessed knowledge can be inspected and interpreted. In this chapter, we shall develop retrieval-augmented modelling ideas further.

Retrieve-and-read setups have been exploited for many years in supervised ODQA setups. REALM (Guu et al., 2020) and ORQA (Lee et al., 2019a) are two recent models that combine pretrained models (Devlin et al., 2019) with a differentiable retriever, and show promising results for ODQA. These models exploit their non-parametric knowledge well. However, their ability to apply their parametric knowledge is limited, since they rely on the span-extractive RC modelling paradigm, constraining them to only produce answers that appear as spans in retrieved documents. This also restricts the task formats they can be applied on. On the other hand, large pretrained sequence-to-sequence (seq2seq) models have recently demonstrated promising performance across a swathe of NLP tasks. A key distinguishing feature of these models is their flexibility, being simple to apply to essentially any NLP problem. However, when applied naively, they are limited to only using parametric knowledge, which, as we have established in chapter 5, is limiting when the task demands precise control over knowledge.

In this chapter, we shall develop a model class that enables us to bring jointly learnable hybrid parametric and non-parametric memory to this modern "workhorse of NLP", i.e. seq2seq models. We shall also broaden our focus beyond simple span-like ODQA in this chapter, and take advantage of the free-form generation capabilities of our formulation to tackle more generalised *knowledge-intensive NLP tasks* (which we defined in section 2.7.3).

Bigger Picture In this chapter, we continue our exploration of ODQA, and how to best represent, store and use knowledge, which we started in the previous chapter. We do, however make a conceptual shift from the first three chapters of this thesis. We shall, for the remainder of this thesis, assume access to a training set of question-answer pairs for the task at hand, in contrast to the previous chapters, which were primarily focused on low-to-no in-domain training data regimes. We do this since our stated aims in the introduction were to understand and improve a number of phenomena in QA modelling, beyond solely the issue data-hungry models. We identified knowledge representation, complexity and inflexibility and slow, expensive inference to be key issues to be improved. This chapter focuses heavily on contributions in terms of knowledge, flexible models and proposes less lossy pipeline models. The following chapters will further contextualise the QA behaviours exhibited by the models we propose in this chapter, and propose an alternative way of storing knowledge, and chapter 8 uses a retrieval-augmented model with a very similar architecture and learning algorithm to that proposed in this chapter. Despite no longer focusing on low data regimes, we note that the findings and techniques from in the first chapters of this thesis should be applicable in combination with the research presented in study in the following chapters. Additional commentary on the connections between this chapter and the wider body of work in the thesis can be found in the conclusion of this chapter (sec. 6.8) and the thesis conclusion, chapter 9.

The material in this chapter first appeared in:

> **Patrick Lewis**, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*

> *Individual Contributions: The initial idea was proposed by a co-author after a discussion between the thesis author and a number of co-authors. The RAG-Token formulation was proposed and implemented as a close collaboration between the thesis author and a co-author. The RAG-Sequence formulation was proposed and implemented by a co-author. The study design, distributed implementation, and all experiments were performed by the thesis author, with the exception of FEVER, which were performed by a co-author. The code was adapted for open-sourcing by a co-author. The original article was written as a close collaboration between several co-authors, with the thesis author writing and editing the majority.*

6.1 Overview

We endow pretrained, parametric-memory seq2seq models with a non-parametric memory through a general-purpose fine-tuning approach which we shall refer to as retrieval-augmented generation (RAG). We build RAG models where the parametric memory is a pretrained seq2seq transformer, and the non-parametric memory is a dense vector index of Wikipedia, accessed with a pretrained dense retriever (introduced in section 2.3.2.2). We combine these components in a probabilistic model, which is fine-tuned end-to-end (Figure 6.1). The retriever, based on Dense Passage Retriever (DPR, Karpukhin et al., 2020) provides latent documents conditioned on the input, and the seq2seq model (BART, Lewis et al., 2020a) then conditions on these latent documents together with the input to generate the output. We marginalise the latent documents with a top-K approximation, either on a per-output basis (assuming the same document is responsible for all tokens) or a per-token basis (where different documents are responsible for different tokens).

Like T5 (Raffel et al., 2020) or BART, RAG can be fine-tuned on any seq2seq task, whereby both the generator and retriever are jointly learned.

There has been extensive previous work proposing architectures to enrich systems with non-parametric memory, which are trained from scratch for *specific tasks*, e.g. memory networks (Weston et al., 2015; Sukhbaatar et al., 2015), stack-augmented networks (Joulin and Mikolov, 2015) and memory layers (Lample et al., 2019). In contrast, in this chapter, we explore a setting where both parametric and non-parametric memory components are pretrained and pre-loaded with extensive knowledge. Crucially, by using pretrained knowledge-access mechanisms, the ability to access knowledge is present without additional training.

Our results re-enforce and build on those from chapter 5, highlighting the benefits of combining parametric and non-parametric memory with generation. In particular, we show the efficacy of this approach for supervised *knowledge-intensive tasks* – tasks that humans could not reasonably be expected to perform without access to an external knowledge source. On four popular ODQA datasets, RAG models strongly outperform both specially pretrained (parametric-memory-only) closed-book QA models (introduced in section 2.4.2.2), and comparable extractive retrieve-and-read (mostly non-parametric-memory) models. For knowledge-intensive free-form generation, we experiment with MSMARCO's NLG task (Bajaj et al., 2016) and Jeopardy question generation, and we find that our models generate responses that are more factual, specific, and diverse than a BART baseline. For FEVER (Thorne et al., 2018) fact verification, RAG achieves results within 4.3% of state-of-the-art pipeline models, which use strong retrieval supervision and specialised architectures. Finally, we demonstrate that the non-parametric memory can be replaced to update the models' knowledge as the world changes.

6.2 Methods

We explore RAG models, which use the input sequence x to retrieve text documents c and use them as additional context when generating the target sequence y. As

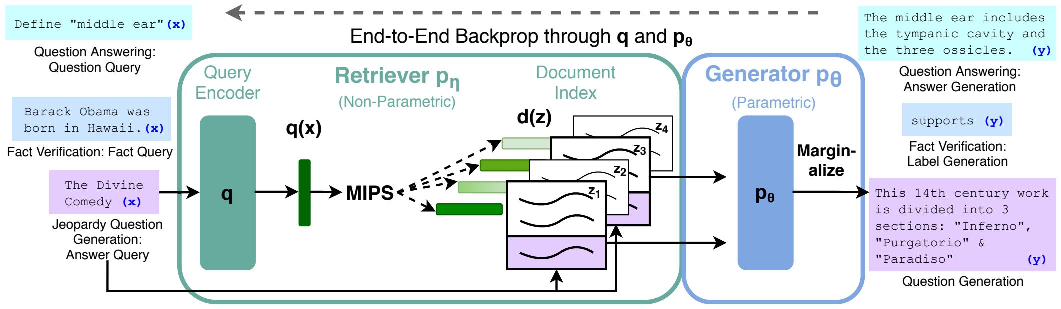

Define "middle ear" (x)
Question Answering: Question Query
Barack Obama was born in Hawaii. (x)
Fact Verification: Fact Query
The Divine Comedy (x)
Jeopardy Question Generation: Answer Query
End-to-End Backprop through q and pθ
Query Encoder
Retriever pη
(Non-Parametric)
Document Index
q
q(x)
MIPS
d(z)
z4
z3
z2
z1
Generator pθ
(Parametric)
pθ
Margin-alize
The middle ear includes the tympanic cavity and the three ossicles. (y)
Question Answering: Answer Generation
supports (y)
Fact Verification: Label Generation
This 14th century work is divided into 3 sections: "Inferno", "Purgatorio" & "Paradiso" (y)
Question Generation

Figure 6.1: RAG models combine a pretrained retriever (*Query Encoder* + *Document Index*) with a pretrained seq2seq model (*Generator*) which are fine-tuned end-to-end. For query x, we use Maximum Inner Product Search to find the top-K documents c. For final prediction y, we treat c as a latent variable and marginalise over seq2seq predictions given different documents.

shown in Figure 6.1, RAG models leverage two components: (i) a retriever $p_\eta(c|x)$ with parameters η that returns (top-K truncated) distributions over text passages given a query x and (ii) a generator $p_\theta(y_i|x,c,y_{1:i-1})$ parametrized by θ that generates the next token based on a context of the previous $i-1$ tokens $y_{1:i-1}$, the original input x and a retrieved passage c.

To train the retriever and generator end-to-end, we treat the retrieved document as a latent variable. We propose two models that marginalise over the latent documents in different ways to produce a distribution over generated text. In one approach, *RAG-Sequence*, documents are marginalised out for complete generated target sequences. The second, *RAG-Token*, marginalises out the documents for every generated token. In the following, we formally introduce both models and then describe the p_η and p_θ components, as well as the training and decoding procedure.

6.2.1 RAG Model Formulations

RAG-Sequence Model The RAG-Sequence model uses the same retrieved document to generate the complete *sequence* before marginalisation. Technically, it treats the retrieved document as a single latent variable that is marginalised to get the seq2seq probability $p(y|x)$ via a top-K approximation. Concretely, the top-K documents are retrieved using the retriever, and the generator produces the output sequence probability for each document, which are then marginalised,

$$p_{\text{RAG-Sequence}}(y|x) \approx \sum_{c\in\text{top-}k(p(\cdot|x))} p_\eta(c|x)p_\theta(y|x,c) = \sum_{c\in\text{top-}k(p(\cdot|x))} p_\eta(c|x)\prod_i^N p_\theta(y_i|x,c,y_{1:i-1})$$

RAG-Token Model In the RAG-Token model we can draw a different latent document for each target *token* and marginalise accordingly. This allows the generator to choose content from several documents when producing an answer. Concretely, the top K documents are retrieved using the retriever, then the generator produces a distribution for the next output token for each document, before marginalising, and repeating the process for the next output token. Formally, we define:

$$p_{\text{RAG-Token}}(y|x) \approx \prod_i^N \sum_{c\in\text{top-}k(p(\cdot|x))} p_\eta(c|x)p_\theta(y_i|x,c_i,y_{1:i-1})$$

Finally, we note that RAG can be used for sequence classification tasks by considering the target class as a sequence of length one, in which case RAG-Sequence and RAG-Token become equivalent.

6.2.2 Retriever: DPR

The retrieval component $p_\eta(c|x)$ is initialised from DPR (Karpukhin et al., 2020). DPR is a typical dense retrieve. A detailed description was given in section 2.3.2.2. To briefly recap, DPR uses a bi-encoder architecture:

$$p_\eta(c|x) \propto \exp\left(\mathbf{d}(c)^\top \mathbf{q}(x)\right) \qquad \mathbf{d}(c) = \text{BERT}_d(z),\ \ \mathbf{q}(x) = \text{BERT}_q(x)$$

where $\mathbf{d}(c)$ and $\mathbf{q}(x)$ are dense representations of a document and a question produced by a $\text{BERT}_{\text{BASE}}$ *document encoder* and *query encoder* respectively. Calculating top-k$(p_\eta(\cdot|x))$, the list of k documents c with highest prior probability $p_\eta(c|x)$, is a Maximum Inner Product Search (MIPS) problem, approximately soluble in sublinear time (Johnson et al., 2019). We use a pretrained bi-encoder to initialise our retriever and build the document index. This retriever was trained to retrieve documents which contain answers to questions in TriviaQA (TQA) and Natural Questions (NQ). We refer to the document index as the *non-parametric memory*.

6.2.3 Generator: BART

The generator component $p_\theta(y_i|x,c,y_{1:i-1})$ could be modelled using any encoder-decoder model. We use BART-large (Lewis et al., 2020a), a pretrained seq2seq transformer (Vaswani et al., 2017). To combine the input x with the retrieved content c when generating from BART, we simply concatenate them. BART was pretrained using a denoising objective and a variety of different noising functions. As we have discussed several times, large pretrained de-noising models store substantial knowledge in its parameters, and we shall refer to BART's generator parameters θ as the *parametric memory* for the remainder of this chapter.

6.2.4 Training

We jointly train the retriever and generator components without any direct supervision on what document should be retrieved. Given a fine-tuning training corpus of J input/output pairs $\mathcal{D} = \{(x_j, y_j)\}_{j=1}^{J}$, we minimise the negative marginal log-likelihood of targets, $\sum_j^J -\log p(y_j|x_j)$ using stochastic gradient descent. Updating the retriever's document encoder BERT$_d$ during training is costly as it requires the document index to be periodically updated. REALM, a similar span-extractive model, requires this during pretraining (Guu et al., 2020). We do not find this step necessary for strong performance, and keep the document encoder (and index) fixed, only updating the query encoder BERT$_q$ and the BART generator.

6.2.5 Decoding

At test time, RAG-Sequence and RAG-Token require different ways to approximate $\arg\max_y p(y|x)$.

RAG-Token RAG-Token can be seen as a standard, auto-regressive seq2seq generator with transition probability:

$$p'_\theta(y_i|x,y_{1:i-1}) = \sum_{c \in \text{top-}k(p(\cdot|x))} p_\eta(c_i|x)p_\theta(y_i|x,c_i,y_{1:i-1})$$

To decode, we can plug $p'_\theta(y_i|x,y_{1:i-1})$ into a standard beam search decoder.

RAG-Sequence For RAG-Sequence, $p(y|x)$ does not break into a conventional per-token likelihood, hence we cannot solve it with simple beam search. Instead, we run beam search for each of the top-k documents, scoring each hypothesis using $p_\theta(y_i|x,c,y_{1:i-1})$. This yields a set of hypotheses $\mathcal{Y}$, some of which may not have appeared in the beams of all documents. To estimate the marginal probability of a hypothesis y , we run an additional generator forward pass for each document c for which y did not already appear in the beam, then multiply each generator probability with $p_\eta(c|x)$ and sum to obtain the marginal. We refer to this decoding procedure as "Thorough Decoding". For longer outputs, $|\mathcal{Y}|$ can become large, requiring many forward passes. For more efficient decoding, we make a further approximation that $p_\theta(y|x,c_i) \approx 0$ if y was not generated during beam search from (x,c_i). This avoids the need to run additional forward passes once the candidate set $\mathcal{Y}$ has been generated. We refer to this decoding procedure as "Fast Decoding".

6.3 Experiments

We experiment with RAG in a wide range of knowledge-intensive tasks, namely standard ODQA tasks, abstractive ODQA, Jeopardy question generation and fact checking. Additional results using RAG for dialogue tasks, entity linking and slot filling can be found in Petroni et al. (2021). For all experiments, we use a single Wikipedia dump for our non-parametric knowledge source, which was described in detail in section 2.5.1. Each article is split into disjoint 100-word documents, to make a total of 21M documents. We build a single MIPS index using FAISS (Johnson et al., 2019) with a Hierarchical Navigable Small World approximation for fast retrieval (Malkov and Yashunin, 2020). During training, we retrieve the top k documents for each query. We consider $k \in \{5,10\}$ for training and set k for test time using dev data. In following, we describe each task in more detail.

6.3.1 Open-Domain Question Answering

Text-based ODQA is the key focus of Parts II and III, and is a prototypical knowledge-intensive NLP task (section 2.7.3). We treat questions and answers as input-output text pairs (x,y) and train RAG by directly minimising the negative log-likelihood of answers. We compare RAG to the popular extractive QA paradigm (Chen et al., 2017; Clark and Gardner, 2018; Lee et al., 2019a; Karpukhin et al., 2020), where answers are extracted spans from retrieved documents, relying primarily on non-parametric knowledge. In particular, we shall compare the retriever-and-read extractive QA model from Karpukhin et al. (2020). This model also uses DPR to retrieve documents, but uses an span-extractive RC model. This model cannot generate answers, only extract, and, as a result, is largely an only-non-parametric-memory model.

We also compare to Closed-Book QA methods (introduced in section 2.4.2.2). We compare to the models of Roberts et al. (2020), which, like RAG, generate answers, but do not use retrieval, instead relying purely on parametric knowledge.

We consider four popular ODQA datasets: NaturalQuestions (NQ), TriviaQA (TQA), WebQuestions (WQ) and CuratedTREC (CT) – see in section 2.5 for detailed descriptions. As CT and WQ are small, we follow DPR (Karpukhin et al., 2020) by initialising CT and WQ models with our NQ RAG model.

6.3.2 Abstractive Question Answering

RAG models can go beyond simple extractive QA and answer questions with free-form, abstractive text generation. To test RAG's natural language generation (NLG) in a knowledge-intensive setting, we use the MSMARCO NLG task v2.1 (Bajaj et al., 2016). The task consists of questions, ten gold passages retrieved from a search engine for each question, and a full sentence answer written by a crowd-worker based on the retrieved passages. We only use the questions and answers and discard the supplied passages, treating MSMARCO as an *open-domain* abstractive QA task. MSMARCO does have some questions that cannot be answered in a way that matches the reference answer without access to the gold passages, such as "What is the weather in Volcano, CA?" so downstream performance will necessarily be lower without using gold passages.[1] We also note that some MSMARCO questions cannot be answered using Wikipedia alone. Here, RAG can rely on para-

[1] We estimate the prevalence of such questions to be $\sim$20%

metric knowledge to generate reasonable responses. We shall evaluate using the standard BLEU-1 and ROUGE-L metric used for MSMARCO's NLG task.

6.3.3 Jeopardy Question Generation

To evaluate RAG's generation abilities to generate something other than answers, but still in a knowledge-intensive setting, we study open-domain question generation. Rather than use questions from standard open-domain QA tasks, which typically consist of short, simple questions, we propose the more demanding task of generating Jeopardy questions. Jeopardy is an unusual format that consists of trying to guess an entity from a fact about that entity. For example, "The World Cup" is the answer to the question "In 1986 Mexico scored as the first country to host this international sports competition twice." As Jeopardy questions are precise, factual statements, generating Jeopardy questions conditioned on their answer entities constitutes a challenging knowledge-intensive generation task.

We use the splits from SearchQA (Dunn et al., 2017). As this is a new task, we train a BART model as a baseline. Following recent practice for evaluating question generation (Zhang and Bansal, 2019), we evaluate using Q-BLEU-1 (Nema and Khapra, 2018). Q-BLEU is a variant of BLEU with a higher weight for matching entities, which correlates more strongly with human judgements.

We also perform two human evaluations: one to assess generation factuality, and one for specificity. We define factuality as whether a statement can be corroborated by trusted external sources, and specificity as high mutual dependence between the input and output (Li et al., 2016). We follow best practice and use pairwise comparative evaluation (Li et al., 2019). Evaluators are shown an answer and two generated questions, one from BART and one from RAG. They are then asked to pick one of four options: question A is better, question B is better, both are good, or neither is good. Further human evaluation details can be found in Appendix F.1

6.3.4 Fact Verification

FEVER (Thorne et al., 2018) is a fact-checking dataset that requires classifying whether a natural language claim is supported or refuted by Wikipedia, or whether there is not enough information to decide. The task requires retrieving evidence from Wikipedia relating to the claim and then reasoning over this evidence to classify whether the claim is true, false, or unverifiable from Wikipedia alone. FEVER is a retrieval problem coupled with an challenging entailment reasoning task. It also provides an appropriate testbed for exploring the RAG models' ability to handle classification rather than generation. We map FEVER class labels (supports, refutes, or not enough info) to single output tokens and directly train with claim-class pairs. Crucially, unlike most other approaches to FEVER, we do not use supervision on retrieved evidence. In many real-world applications, retrieval supervision signals aren't available, and models that do not require such supervision will be applicable to a wider range of tasks. We explore two variants: the standard 3-way classification task (supports/refutes/not enough info) and the 2-way (supports/refutes) task studied in Thorne and Vlachos (2021). In both cases we report label accuracy.

6.4 Implementation Details

We train all RAG models and BART baselines using Fairseq (Ott et al., 2019). We train with mixed precision floating point arithmetic (Micikevicius et al., 2018), distributing training across 8 NVIDIA 32GB V100 GPUs, although training and inference can be accomplished on one GPU. We find that MIPS with FAISS is sufficiently fast on CPU, so we store document index vectors on CPU, requiring $\sim$ 36 GB of CPU memory for all of Wikipedia when employing SQ8-quantization. Further reduction to index size would be straightforward, using approaches such as Product Quantization (Jégou et al., 2011).

Hyper-parameters are set using development data. For Open-domain QA we report test numbers using 15 retrieved documents for RAG-Token. For RAG-Sequence, we report test results using 50 retrieved documents, and we use thorough decoding since answers are generally short. We use greedy decoding for ODQA as we did not

Model		NQ	TQA	WQ	CT
Closed Book	T5-11B (Roberts et al., 2020)	32.6	42.3	37.2	-
	T5-11B+SSM (Roberts et al., 2020)	34.8	51.0	40.8	-
Open Book	REALM (Guu et al., 2020)	40.4	-	40.7	46.8
	DPR + RC (Karpukhin et al., 2020)	41.5	**56.8**	42.4	49.4
RAG	RAG-Token	44.1	55.2	**45.5**	50.0
	RAG-Sequence	**44.5**	**56.8**	45.2	**52.2**
Subsequent State-of-the-Art (Izacard and Grave, 2021b)		*51.4*	*67.3*	-	-

Table 6.1: ODQA Test EM Scores for closed-book, open-book and RAG. RAG outperforms only-parametric-memory T5, and the predominately non-parametric-memory DPR+RC retrieve-and-read. RAG represented the state-of-the-art at the time these experiments were performed. A current state-of-the-art model, “Fusion-in-Decoder”, which has a similar philosophy – but different architecture – to RAG, is shown for context.

Model	Jeopardy		MSMARCO		FVR3	FVR2
	BLEU-1	QBLEU-1	ROUGE-L	BLEU-1	Label Acc.	
Representative SotA	-	-	**49.8***	**49.9***	**76.8**	**92.2***
BART	15.1	19.7	38.2	41.6	64.0	81.1
RAG-Token	**17.3**	**22.2**	40.1	41.5	72.5	89.5
RAG-Sequence	14.7	21.4	40.8	44.2		

Table 6.2: Generation and classification Test Scores. MSMARCO SotA is Bi et al. (2020), FEVER-3 is Zhong et al. (2020) and FEVER-2 is Thorne and Vlachos (2021) *Uses gold context/evidence. Best model without gold access underlined.

find beam search improved results. For Open-MSMARCO and Jeopardy question generation, we report test numbers using ten retrieved documents for both RAG-Token and RAG-Sequence, and we also train a BART-large model as a baseline. We use a beam size of four, and use the “fast decoding” approach for RAG-Sequence, as “thorough decoding” did not improve performance.

6.5 Results

6.5.1 Open-Domain Question Answering Results

Table 6.1 shows RAG strongly outperforms the competing closed-book and open-book approaches available when these experiments were performed. RAG combines the generation flexibility of the “closed-book” (parametric-only) approaches and the performance of ”open-book” retrieval-based approaches. Unlike REALM and T5+SSM, RAG enjoys strong results without expensive, specialised “salient span masking” pretraining (Guu et al., 2020). It is worth noting that RAG’s retriever is initialised using DPR’s retriever, which does use retrieval supervision on Natural Questions and TriviaQA. RAG compares favourably to the DPR+RC ODQA model, which uses a BERT-based “cross-encoder” to re-rank documents, along with an extractive reader. RAG demonstrates that neither an explicit re-ranker nor extractive RC reader model is necessary for strong ODQA performance.

There are several advantages to generating answers even when it is possible to extract them. Documents with clues about the answer but do not contain the answer verbatim can still contribute towards a correct answer being generated, which is not possible with standard extractive approaches, leading to more effective marginalisation over documents. Furthermore, RAG can generate correct answers even when the correct answer is not in any retrieved document, achieving 11.8% accuracy in such cases for NQ, where an extractive model would score 0%.

6.5.2 Abstractive Question Answering Results

As shown in Table 6.2, RAG-Sequence outperforms BART on Open MSMARCO NLG by 2.6 BLEU points and 2.6 ROUGE-L points. RAG approaches contemporary state-of-the-art methods, which is remarkable given that (i) those models access gold passages with specific information required to generate the reference answer, (ii) many questions are unanswerable without the gold passages, and (iii) not all questions are answerable from Wikipedia alone. Table 6.3 shows some generated answers from our models. Qualitatively, we find that RAG models hallucinate less and generate factually correct text more often than BART. Later, we also show that RAG generations are more diverse than BART generations (see section 6.5.5).

6.5.3 Jeopardy Question Generation Results

Table 6.2 shows that RAG-Token performs better than RAG-Sequence on Jeopardy question generation, with both models outperforming BART on Q-BLEU-1. Table 6.4 shows human evaluation results, over 452 pairs of generations from BART and RAG-Token. Evaluators indicated that BART was more factual than RAG in only 7.1% of cases, while RAG was more factual in 42.7% of cases, and both RAG and BART were factual in a further 17% of cases, clearly demonstrating the effectiveness of RAG on the task over a state-of-the-art generator. Evaluators also find RAG generations to be more specific by a large margin. Table 6.3 shows typical generations from each model.

Jeopardy questions often contain two separate pieces of information, and RAG-Token may perform best because it can generate responses that combine content from several documents. Figure 6.2 shows an example. When generating “Sun”, the posterior is high for document 2 which mentions “The Sun Also Rises”. Similarly, document 1 dominates the posterior when “A Farewell to Arms” is generated. Intriguingly, after the first token of each book title is generated, the document posterior flattens. This observation suggests that the generator can complete the titles without depending on specific documents. In other words, the model’s parametric knowledge is sufficient to complete the titles.

We find evidence for this hypothesis by feeding the BART-only baseline with the partial decoding “The Sun. BART completes the generation “The Sun Also Rises” is a novel by this author of “The Sun Also Rises” indicating the title “The Sun Also Rises” is stored in BART’s parameters. Similarly, BART will complete the partial decoding “The Sun Also Rises” is a novel by this author of ”A with “The Sun Also Rises” is a novel by this author of “A Farewell to Arms”. This example shows how parametric and non-parametric memories *work together* – the non-parametric component helps guide generation, drawing out specific knowledge stored in the parametric memory.

6.5.4 Fact Verification Results

Table 6.2 shows our results on FEVER. For 3-way classification, RAG scores are within 4.3% of state-of-the-art models, which are complex pipeline systems with domain-specific architectures and substantial engineering, trained using intermediate retrieval supervision, which RAG does not require. For 2-way classification, we compare against Thorne and Vlachos (2021), who train RoBERTa (Liu et al.,

Document 1: his works are considered classics of American literature [...] His wartime experiences formed the basis for his novel "**A Farewell to Arms**"(1929)
Document 2: [...] artists of the 1920s "Lost Generation" expatriate community. His debut novel "**The Sun Also Rises**", was published in 1926.

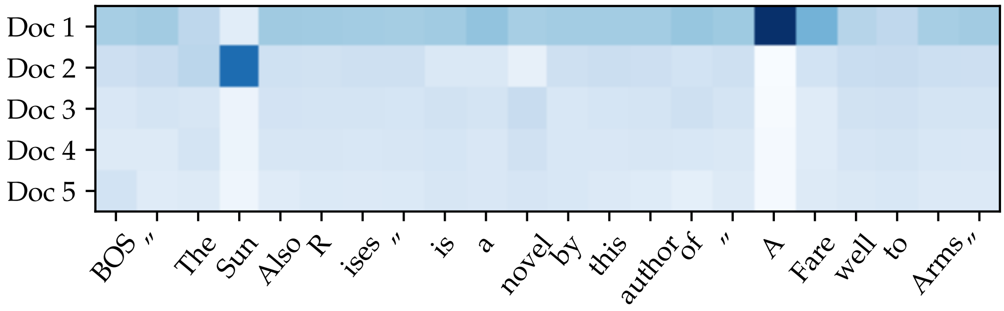


Figure 6.2: RAG-Token document posterior $p(c_i|x, y_i, y_{-i})$ for each generated token for input "Hemingway" for Jeopardy generation with 5 retrieved documents. The posterior for document 1 is high when generating "A Farewell to Arms" and for document 2 when generating "The Sun Also Rises".

	Input	Model	Generation
MSMARCO	define middle ear	BART	$^?$The middle ear is the part of the ear between the middle ear and the nose.
		RAG-T	The middle ear is the portion of the ear internal to the eardrum.
		RAG-S	The middle ear includes the tympanic cavity and the three ossicles.
	what currency needed in scotland	BART	The currency needed in Scotland is Pound sterling.
		RAG-T	Pound is the currency needed in Scotland.
		RAG-S	The currency needed in Scotland is the pound sterling.
Jeopardy Q-Gen	Washington	BART	$^?$This state has the largest number of counties in the U.S.
		RAG-T	It's the only U.S. state named for a U.S. president
		RAG-S	It's the state where you'll find Mount Rainier National Park
	The Divine Comedy	BART	*This poem by Dante is divided into 3 parts: the Inferno, the Purgatorio & the Purgatorio
		RAG-T	Dante's "Inferno" is the first part of this epic poem
		RAG-S	This 14th century work is divided into 3 sections: "Inferno", "Purgatorio" & "Paradiso"

Table 6.3: Examples from generation tasks. RAG models generate more specific and factually accurate responses. '?' indicates factually incorrect responses, * indicates partially correct responses.

2019c) to classify the claim as true or false given the gold evidence sentence. RAG achieves an accuracy within 2.7% of this model, despite being supplied with only the claim and retrieving its own evidence. We also analyse whether documents retrieved by RAG correspond to documents annotated as gold evidence in FEVER. We find that the top retrieved document is from a gold article in 71% of cases, and a gold article is present in the top 10 retrieved articles in 90% of cases.

	Factuality	Specificity
BART better	7.1%	16.8%
RAG better	**42.7%**	**37.4%**
Both good	11.7%	11.8%
Both poor	17.7%	6.9%
No majority	20.8%	20.1%

Table 6.4: Human assessments for Jeopardy Question Generation

	MSMARCO	Jeopardy Q-Gen
Gold	89.6%	90.0%
BART	70.7%	32.4%
RAG-Token	77.8%	46.8%
RAG-Seq.	83.5%	53.8%

Table 6.5: Ratio of distinct to total tri-grams for generation tasks

	NQ	TQA	WQ	CT	Jeopardy Q-Gen		MSMARCO		FVR-3	FVR-2
		Exact Match			B-1	QB-1	R-L	B-1	Label Accuracy	
RAG-Tok (BM25)	29.7	41.5	32.1	33.1	17.5	22.3	55.5	48.4	**75.1**	**91.6**
RAG-Seq (BM25)	31.8	44.1	36.6	33.8	11.1	19.5	56.5	46.9		
RAG-Tok (Frozen)	37.8	50.1	37.1	51.1	16.7	21.7	55.9	49.4	72.9	89.4
RAG-Seq (Frozen)	41.2	52.1	41.8	52.6	11.8	19.6	56.7	47.3		
RAG-Tok (end2end)	43.5	54.8	**46.5**	51.9	**17.9**	**22.6**	56.2	**49.4**	74.5	90.6
RAG-Seq (end2end)	**44.0**	**55.8**	44.9	**53.4**	15.3	21.5	**57.2**	47.5		

Table 6.6: Dev set Ablations for different retrievers. BM25 indicates BM25 was used rather than a dense retriever. "end2end" and "Frozen" indicates the dense retriever was trained end-to-end, or kept frozen respectively. As FEVER is a classification task, both RAG models are equivalent.

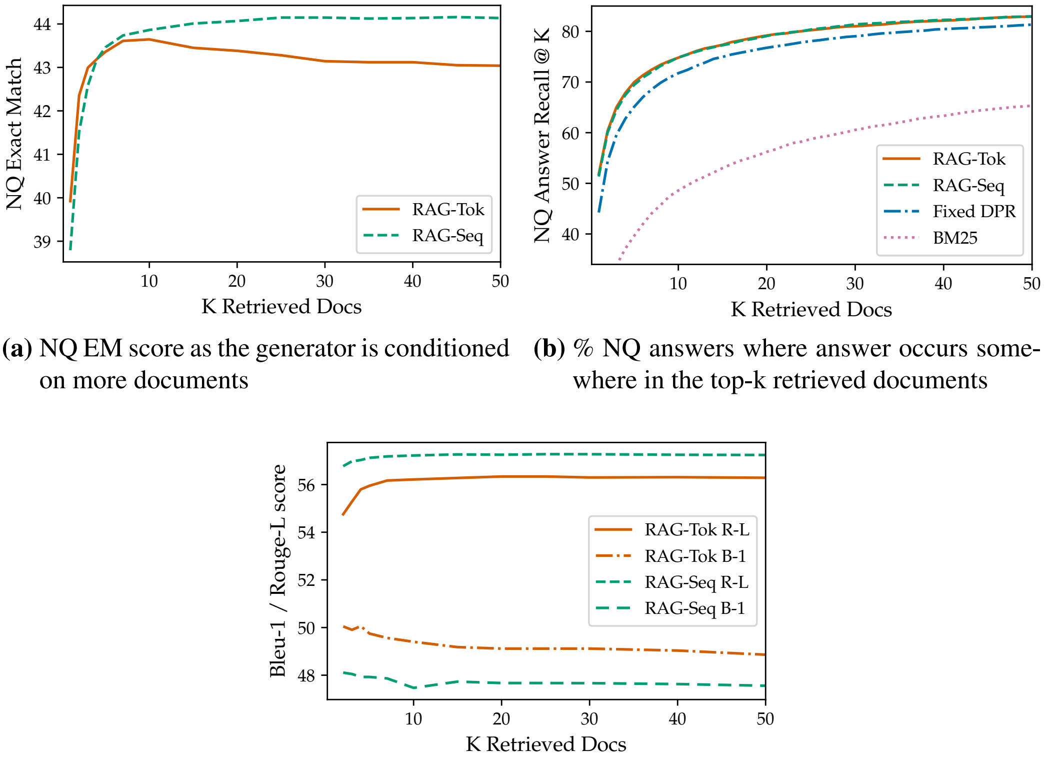
6.5.5 Additional Results

Generation Diversity We saw in section 6.5.3 that RAG models are more factual and specific than BART. Following recent work on diversity-promoting decoding (Li et al., 2016; Vijayakumar et al., 2018; Massarelli et al., 2020), we also investigate generation diversity by calculating the ratio of distinct ngrams to total ngrams generated by different models. Table 6.5 shows that RAG-Sequence's generations are more diverse than RAG-Token's, and both are significantly more diverse than BART without needing any diversity-promoting decoding schemes.

Retrieval Ablations A key feature of RAG is learning to retrieve relevant information for the task at hand. To assess the effectiveness of the learnt retrieval mechanism, we run ablations where we do not optimise the retriever during training. As shown in Table 6.6, learned retrieval improves results for all tasks. Figure 6.3b demonstrates that the learned retriever shows a higher recall for gold documents compared to the fixed DPR retriever. We also compare RAG's dense retriever to a word overlap-based BM25 retriever (Robertson and Zaragoza, 2009). Here, we replace RAG's retriever with a fixed BM25 system, and use BM25 retrieval scores as logits when calculating $p(c|x)$. Table 6.6 show the results. For FEVER, BM25 performs best, perhaps since FEVER claims are heavily entity-centric and thus well-suited for word overlap-based retrieval. TF-IDF was also used in aspects of the construction of the FEVER dataset, possibly hinting at SQuAD-style biases (section 2.5.2.2). Differentiable retrieval improves results on all other tasks, especially for ODQA, where it is crucial.

Index hot-swapping An advantage of non-parametric memory models like RAG is that knowledge can be easily updated at test time. Parametric-only models would need further training to update their behaviour as the world changes, and updating their knowledge without catastrophic forgetting is an open research problem. To demonstrate, we build an index using a Wikipedia dump from December 2016 and compare outputs from RAG using this index to the newer index from our main results (December 2018). We prepare a list of 82 world leaders who had changed between these dates and use a template "Who is {position}?" (e.g. "Who is the President of Peru?") to query our NQ RAG model with each index. RAG answers 70% correctly using the 2016 index for 2016 world leaders and 68% using the 2018 index for 2018 world leaders. Accuracy with mismatched indices is low (12% with the 2018 index for 2016 leaders, 4% with the 2016 index for 2018 leaders). This shows we can update RAG's world knowledge by simply replacing its non-parametric memory.

Effect of Retrieving More Documents Models are trained with either 5 or 10 retrieved latent documents, and we do not observe significant differences in performance between them. We have the flexibility to adjust the number of retrieved documents at test time, to tune accuracy vs. speed. Figure 6.3a shows that retrieving more documents at test time monotonically improves ODQA for RAG-Sequence, but performance peaks for RAG-Token at 10 retrieved documents. Figure 6.3c shows that retrieving more documents leads to higher ROUGE-L for RAG-Token at the expense of BLEU-1, but the effect is less pronounced for RAG-Sequence.



(a) NQ EM score as the generator is conditioned on more documents

(b) % NQ answers where answer occurs somewhere in the top-k retrieved documents

(c) MSMARCO BLEU-1 and ROUGE-L as more documents are retrieved

Figure 6.3: How test-time performance changes as the number of documents used to condition the RAG model is increased.

6.6 Additional Observations and Negative Results

Retrieval Collapse In preliminary experiments, we observed that for some tasks such as story generation (Fan et al., 2018), the retrieval component would "collapse" and learn to retrieve the same documents regardless of the input. In these cases, once retrieval had collapsed, the generator would learn to ignore the documents, and the RAG model would perform equivalently to BART. The collapse is likely due to a kind of cold-start problem: if the retriever doesn't surface useful documents in the top-K for generating the output at the beginning of training, then the generator may learn to condition less on documents, which in turn may result in less informative gradients for the retriever, exacerbating the problem. Improving the stability of the retriever to this kind of issue, perhaps by artificially including some known useful documents in the top-K at the beginning of training, is important future work. Perez et al. (2019) also found spurious retrieval results when optimising a retrieval component in order to improve performance on downstream tasks.

Null-Document Probabilities We experimented with adding a “Null document” mechanism to RAG, similar to REALM (Guu et al., 2020) in order to model cases where no useful information could be retrieved for a given input. Here, if k documents were retrieved, we would additionally “retrieve” an empty document and predict a logit for the null document, before marginalising over $k+1$ predictions. We explored modelling this null document logit by learning (i) a document embedding for the null document, (ii) a static learnt bias term, or (iii) a neural network to predict the logit. We did not find that these improved performance, so in the interest of simplicity, we omit them. For the MSMARCO task, where useful retrieved documents cannot always be retrieved, we observe that the model learns to always retrieve a particular set of documents when the question is less likely to benefit from retrieval, suggesting that model learns its own latent strategy for dealing with questions that don’t have evidence.

6.7 Related Work

Here we shall highlight some related work that has not already been covered in detail in previous chapters.

Single-Task Retrieval Prior work has shown that retrieval improves performance across a variety of NLP tasks when considered in isolation. Other than ODQA, which we have discussed in detail in chapter 2, work in fact checking (Thorne et al., 2018), fact completion (Petroni et al., 2020), long-form question answering (Fan et al., 2019), Wikipedia article generation (Liu et al., 2018), dialogue (Moghe et al., 2018; Weston et al., 2018; Dinan et al., 2019; Fan et al., 2021), translation (Gu et al., 2018), and language modelling (Guu et al., 2018; Khandelwal et al., 2020) has shown improvements by employing retrieval. Our contribution in this chapter to is unify previous successes in incorporating retrieval for individual tasks, showing that a single retrieval-based architecture is capable of achieving strong performance across several knowledge intensive tasks.

General-Purpose Architectures for NLP Prior work on general-purpose architectures for NLP tasks has shown great success without the use of retrieval. A single, pretrained language model has been shown to achieve strong performance on various classification tasks in the GLUE benchmarks (Wang et al., 2018a, 2019) after fine-tuning (Radford et al., 2018; Devlin et al., 2019). GPT-2 (Radford et al., 2019) later showed that a single, left-to-right, pretrained language model could achieve strong performance across both discriminative and generative tasks. For further improvement, BART (Lewis et al., 2020a) and T5 (Raffel et al., 2020; Roberts et al., 2020) propose a single, pretrained encoder-decoder model that leverages bidirectional attention to achieve stronger performance on discriminative and generative tasks. Our work aims to expand the space of possible tasks which can be tacked by a single, unified architecture.

Learned Retrieval There is significant work on learning to retrieve documents in information retrieval, more recently with pretrained, neural language models (Nogueira and Cho, 2019; Karpukhin et al., 2020) similar to ours. Some work optimises the retrieval module to aid in a specific, downstream task such as question answering, using search (Perez et al., 2019), reinforcement learning (Choi et al., 2017; Wang et al., 2018d,c), or a latent variable approach (Lee et al., 2019a; Guu et al., 2020) as in our work. These successes leverage different retrieval-based architectures and optimisation techniques to achieve strong performance on a single task, while we show that a single retrieval-based architecture can be fine-tuned for strong performance on a variety of tasks.

Memory-based Architectures Our document index can be seen as a large external memory for neural networks to attend to, analogous to memory networks (Weston et al., 2015; Sukhbaatar et al., 2015). Concurrent work (Févry et al., 2020) learns to retrieve a trained embedding for each entity in the input, rather than to retrieve raw text as in our work. Other work improves the ability of dialog models to generate factual text by attending over fact embeddings (Dinan et al., 2019; Fan et al., 2021) or, closer to our work, over retrieved text directly (Ghazvininejad et al., 2018). A key feature of our memory is that it is comprised of raw text rather than distributed representations, which makes the memory both (i) human-readable, lending a form of interpretability and provenance to our model, and (ii) human-writable, enabling us to dynamically update the model's memory by editing the document index.

Retrieve-and-Edit approaches Our method shares some similarities with retrieve-and-edit style approaches, where a similar training input-output pair is retrieved for a given input, and then edited to provide a final output. These approaches have proved successful in a number of domains including Machine Translation (Gu et al., 2018; Hossain et al., 2020) and Semantic Parsing (Hashimoto et al., 2018). Our approach has several differences, including less of emphasis on lightly editing a retrieved item, but on aggregating content from several pieces of retrieved content, as well as learning latent retrieval, and retrieving evidence documents, rather than related training pairs. This said, RAG techniques may work well in these settings, and could represent promising future work.

6.8 Conclusion

In this chapter, we presented hybrid seq2seq models with access to parametric and non-parametric memory. We showed that our RAG models strongly outperformed contemporary comparable models on ODQA. We found that people prefer RAG's generation over a purely parametric generator, finding RAG more factual and specific. We conducted a thorough investigation of the learned retrieval component, validating its effectiveness, and we illustrated how the retrieval index can be hot-swapped to update the model without requiring any retraining. Our work opens up new research directions on how parametric and non-parametric memories interact, how to best combine them for a wide variety of NLP tasks.

It is worth highlighting a few limitations and open directions in retrieval-augmented generators. We noted that RAG can occasionally suffer from "retrieval-collapse" when the retriever initially cannot surface any useful documents for the task at hand. This could be overcome by using a specialised retriever initialisation, but this relies on some retrieval supervision data, or using other heuristics which can be applied to help with initial relevance. Development of stronger general-purpose ad-hoc retrievers may help here, as may innovations in how RAG models are trained.

A second limitation is that whilst we have demonstrated that RAG models are capable of combining or *fusing* information from several passages via marginalisation, this interaction happens "late" in the forward pass - i.e. after all neural processing. Subsequent work has shown that conditioning the generator jointly on a *set* of documents, rather than on each document separately, and fusing information across documents *before* feeding into the generator's decoder, is more empirically effective. One such approach, called Fusion-in-Decoder (FiD, Izacard and Grave, 2021b) has demonstrated very strong results for ODQA (included for reference in table 6.1. In the remainder of this thesis, we shall make extensive use of FiD in our studies of ODQA. Conditioning this way complicates end-to-end learning, and as such, FiD is a pipe-lined model, relying on a fixed retriever. This can be mitigated using a kind of block descent approach (Izacard and Grave, 2021a) and there is promising work from Sachan et al. (2021) demonstrating an end-to-end approach. In future work, it may be fruitful to investigate if the two components can be jointly pretrained, either with a denoising objective similar to BART or some another objective.

To conclude Part II, we have developed retrieval-augmented hybrid parametric/non-parametric models – first, in chapter 5 in an unsupervised setting, showing its effectiveness on a knowledge-probing task, and then in chapter 6 in a supervised generation setting, showing strong results on ODQA and related tasks. In Part III, we take a deeper look into *where* the knowledge required to answer ODQA questions can be found, and hence, develop new strategies about what forms to store it in, and whether these forms are best suited for parametric or non-parametric media.

Part III

Memorisation and Generalisation in Open-Domain QA

“I’m sorry, my responses are limited. You must ask the right questions.”

Dr Lanning’s Hologram

I, Robot (Movie)

Chapter 7

Question And Answer Test-Train Overlap in Open-Domain QA Datasets

In Part II, we developed retrieval-augmented models which could combine parametric knowledge and non-parametric knowledge, and showed how they could be applied to tasks like ODQA. In particular, in chapter 6, we used supervised ODQA tasks to benchmark how well systems could command knowledge expressed in a textual knowledge source. We are not alone in using these datasets in this way – using supervised ODQA tasks for this purpose has become relatively standard in knowledge-intensive modelling (Guu et al., 2020; Roberts et al., 2020; Févry et al., 2020; Verga et al., 2020; Petroni et al., 2021; Min et al., 2021).

However, so far, we have not really developed a deep understanding of *where* the knowledge required to answer the questions in these datasets can be found. Whilst there have been several works examining other kinds of QA datasets (Manjunatha et al., 2019; Kaushik and Lipton, 2018; Sugawara et al., 2018, 2020), we know comparatively little about how the questions and answers are distributed in these ODQA benchmarks. As a result, we do not have a good understanding of how our models come up with answers, and what kinds of knowledge they actually store and access in their parametric and non-parametric knowledge mechanisms.

In Part III, we will address this issue in detail, starting, in this chapter, with deep analysis of the test sets of three popular ODQA datasets, namely WebQuestions (WQ), TriviaQA (TQA) and NaturalQuestions (NQ). Detailed descriptions of these datasets can be found in section 2.5. Our analysis will be behavioural in focus, and we shall deeply investigate the extent to which memorisation and generalisation are required for these ODQA benchmarks, and, in so doing, gain greater insights into how models equipped with different knowledge mechanisms actually operate.

Bigger Picture This chapter asks a key, yet-unanswered question, highly relevant to the task studied in the previous chapters. We have ascertained in chapters 5 and 6 that retrieval-augmentation is important for answering questions well, but we haven't yet addressed what knowledge we should be retrieving from. This chapter will look at this question through the lens of a behavioural analysis, and will make the following observation: often, a lot knowledge required for ODQA tasks is present in the training data, and so directly retrieving from training data, i.e. QA pairs, (rather than a background corpus of text documents, as we have done before) is useful and insightful. A model that exploits this observation will be introduced in this chapter which will form the foundation of the next (and final) chapter, chapter 8. Finally, The annotation procedure which we shall develop in this chapter builds on the ideas from chapter 4, where we used a retrieval/matching technique to reduce annotation workload, a strategy we shall employ again here for a different end goal. Additional commentary on the connections between this chapter and the wider body of work in the thesis can be found in the conclusion of this chapter (sec. 7.6) and the thesis conclusion, chapter 9.

The material in this chapter first appeared in:

> **Patrick Lewis**, Pontus Stenetorp, and Sebastian Riedel. 2021. Question and Answer Test-Train Overlap in Open-Domain Question Answering Datasets. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*. Best Paper Award Recipient.

Individual Contributions: The initial observations of overlap and memorisation was made by the thesis author. The experimental design, experiments and analysis were devised by the thesis author, with advice from co-authors. Annotation was shared between all authors. The majority of the original article was written by the thesis author, with feedback by co-authors.

7.1 Overview

We identify three classes of question that a trained ODQA system should be able to answer, in increasing order of difficulty:

1. The most basic behaviour is to be able to reliably recall the answer to a question that the model has seen at training time.

2. A model should be able to answer novel questions at test time and choose an answer from the set of answers it has seen during training.

3. a strong system should be able to answer novel questions which have answers which are not contained in the training data.

It is not clear to what extent our current ODQA datasets measure each of these three behaviours. To address this, we stratify the test sets of these datasets. Firstly, we split the test data by whether answers in the test set also appear somewhere in the training sets. We find that 58-71% of test answers also occur somewhere in the training data, demonstrating that the majority of the test data does not require generalising to unseen answers.

Secondly, we annotate 1000 question-answer pairs from each test set for repeated questions in their respective training sets. We find that a surprisingly high 28-34% of test questions have paraphrased questions in the training data, the vast majority of which are near-duplicates differing by one or two words. This result implies that 30% of the test set of these datasets only probe for how well models can simply memorise question-answer pairs seen at training.

Equipped with these insights, we compute the performance of several ODQA mod-

Dataset	% with Answer Overlap	% with Question Overlap
Natural Questions	63.6	32.5
TriviaQA	71.7	33.6
WebQuestions	57.9	27.5

Table 7.1: Fractions of open-domain test sets that overlap with their training sets.

els on our test subsets, including both open-book retrieve-and-read and closed-book models. We find that test instances with train-overlapping data contribute the bulk of the overall performance of all the models studied.

These issues seem to be more acute for closed-book models. Strikingly, we find that a closed-book model based on BART (Lewis et al., 2020a) is incapable of producing answers not observed at training time, and achieves very low scores on non-overlapping questions, suggesting this model is only capable of memorising question-answer pairs from training time. With this in mind, we build simple nearest-neighbour models which outperform this BART model, despite having virtually no capacity to generalise beyond training data. We refer to these simple nearest-neighbour models as *Question-Answer-Pair (QA-pair) retrievers*, and they form much of the modelling basis for Part III. In this chapter, we demonstrate their competitive performance with closed-book QA, and in chapter 8, we shall greatly expand their capabilities.

To summarise, we make the following contributions: 1) We provide insights into how answer entities are distributed between dataset splits for ODQA datasets 2) We provide annotated subsets of ODQA test sets indicating whether test-time questions are duplicates of training time questions. 3) We evaluate a variety of models on our dataset splits, and derive insights into what kinds of question answering behaviour different models achieve.

7.2 Datasets

In our analysis, we consider three widely used ODQA datasets: WebQuestions (Berant et al., 2013), TriviaQA (Joshi et al., 2017), and NaturalQuestions, a subset of NaturalQuestions (Kwiatkowski et al., 2019) introduced by Lee et al. (2019a). All

NaturalQuestions		TriviaQA		WebQuestions	
Overlapping	Non-overlapping	Overlapping	Non-overlapping	Overlapping	Non-overlapping
Phil Simms	Cloves	David Bowie	Death in the afternoon	Harvard	Queen Victoria
Brian Johnson	Matt Monro	Battle of camlann	Clash of the Titans	Alderaan	Brasília
8	1,020-1,080 kg	Heligoland	ice-cream sundae	India	Paddington
the Indians	Hermann Ebbinghaus	Henry VII	Camshaft	2011	Tom Corbett
the 1830s	Matt Flinders	Niagra Falls	Cumberland	Zeus	Gary

Table 7.2: Randomly sampled overlapping and non-overlapping answers from all test sets.

three datasets consist of factual natural language questions and short multi-token answers, but differ slightly in the style of questions and format of answers. These datasets, and their evaluation metrics are described in detail in chapter 2.

For all three datasets, the canonical train, development and test splits were obtained by randomly splitting the data, and there are no exact duplicate questions in any dataset. We exclude development data from our analyses, focusing purely on train-test overlap to explicitly assess the effects of training memorisation.

7.3 Test-Train Overlaps

We explore two ways of examining the test sets based on overlaps between training and test data. We assume there is a set of question-answer pairs, $\mathcal{D}_{\text{all}} = \left\{\left(q_j, a_j\right)\right\}_{j=1}^{J}$ which has been partitioned into a test set $\mathcal{D}_{\text{test}} = \{(q_{\text{ts},i}, a_{\text{ts},i})\}_{i=1}^{M}$ and a training set $\mathcal{D}_{\text{train}} = \left\{\left(q_{\text{tr},k}, a_{\text{tr},k}\right)\right\}_{k=1}^{N}$. Consider a question-answer pair $(q_{\text{ts}}, a_{\text{ts}})$ from $\mathcal{D}_{\text{test}}$ where the answer consists of a set of at least one answer reference $a_{\text{ts}} = \{s_1..s_n\}$. We can define *answer overlap* to be where there exists at least one $(q'_{\text{tr}}, a'_{\text{tr}}) \in \mathcal{D}_{\text{train}}$ which shares at least one answer reference with $(q_{\text{ts}}, a_{\text{ts}})$. We can also define *question overlap* to be where there exists some $(q''_{\text{tr}}, a''_{\text{tr}}) \in \mathcal{D}_{\text{train}}$ where q''_{tr} is a paraphrase of q_{ts} *and* a''_{ts} shares at least one answer reference with a_{ts}.

Answer Overlap Following Rajpurkar et al. (2016), we apply answer normalisation[1] on answer references before searching for overlapping answer references for all QA-pairs in the test set – see Table 7.1. We find that 58% of test QA-pairs in WebQuestions have answer overlaps, with 63.6% and 71.7% for NQ and TriviaQA respectively. We would naturally expect TriviaQA to have higher answer overlap as

[1] Answer normalisation consists of lower-casing, stripping punctuation, removing articles and normalising whitespace

#	Answer	Test Question	Train Question
1	Jason Marsden	who plays max voice in a goofy movie	who does max voice in a goofy movie
2	January 23 2018	when will the 2018 oscar nominations be announced	when are the oscar nominations for 2018 announced
3	Alan Shearer	who has scored more goals in the premier league	most goals scored by a premier league player
4	retina	where are the cones in the eye located	where are cone cells located in the eye
5	Francisco Pizarro	who led the conquest of the incas in south america	conquistador who defeated the incan empire in peru

Table 7.3: Randomly sampled test-train overlapping questions in NQ. See Appendix G.1 for more examples, including examples from TriviaQA and WebQuestions

it has more answer references per question on average (13.7 references on average compared to 1.2 for NQ and 2.4 for WebQuestions). Examples of answer overlaps are shown in Table 7.2.

Question Overlap Unlike answer overlap, question overlap cannot be trivially computed automatically, as searching for duplicates via rules or paraphrase classifiers may lead to both false positives and negatives. Thus, we turn to manual annotation to investigate question overlap. To obtain a representative sample for each dataset, we annotate a random subset of 1,000 QA-pairs for each test set. Annotators are shown a list of up to 50 training questions which have a similar answer reference. Training questions are selected for annotation if one of the following is true: they share an answer reference with a test question, a test answer reference is a sub-sequence of a training answer reference, or the other way around (a training reference answer is a sub-sequence of a test answer reference). If there are more than 50 such questions, the top 50 are chosen by the highest degree of word overlap to the test question. This answer similarity function is designed for high recall to obtain a tight lower bound on question overlap. If there were no questions with similar answers in the training set, the question was automatically annotated as not overlapping. Three expert annotators looked through these similar questions and indicated if any were paraphrases of the test question and had the same answer.

The results from the annotation can be seen in Table 7.1 and examples of overlapping questions in Table 7.3. A sample of 100 2-way annotated examples indicated 93% agreement, corresponding to a Cohen's Kappa of 0.85 (Cohen, 1960). What we observe is a high degree of question overlap, with between 27.5 and 33.6% of the 1,000 annotated test questions having a duplicate in the training set. It is also common to see several duplicates per test question, with an average of 2.8 duplicate questions per overlapping test question in NQ.

7.4 Implications for Modelling

Given our findings from above, we turn our attention to how well ODQA models perform with respect to train-test set overlap. Earlier, we identified three classes of answering behaviours: 1) questions that can be memorised at training time, 2) novel questions that can be answered with answers memorised at training time, 3) novel questions with novel answers. We refer to these behaviours as *Question memorisation*, *Answer classification* and *QA generalisation* respectively.

Question Memorisation To perform well on the question overlap subset, a model would only need to be able to memorise QA-pairs at training time, and then recognise which training question matches a test-time question. The reasoning required ranges from trivial duplicate detection for very similar questions such as "who played pink in pink floyd the wall" and "who played pink in the movie the wall", to more challenging inference problems for more subtle duplicates such as "On which island in the North Sea did both St Aidan and St Cuthbert live?" and "irish born missionary saint aidan founded a monastery in 653 on which english island which is also the name of a 1970s uk folk-rock band?". A manual annotation of 100 question-overlap pairs indicated that 81% were simple duplicates differing by one or two words, 14% required some paraphrasing recognition capability, and 5% required more sophisticated natural language understanding. To measure performance on question memorisation, we build a test subset comprised of QA-pairs which have question overlap to the training set, which we refer to as the *question overlap test subset*.

Answer Classification In order to tackle the answer-overlap question, a multi-class classifier over training set answers would be sufficient, as answers never appear at test time that don't appear at training time. We build a test subset of QA-pairs which have answer overlap, but do not have question overlap. Question-overlap pairs are excluded to isolate performance on answer classification, since question-overlap questions are significantly easier to answer, and would inflate scores. We refer to this test subset at the *answer-only overlap test subset*, and will use it to measure performance on the answer classification competency.

QA Generalisation In this regime, models cannot rely on memorising their training data. To measure performance on this most challenging split, we build a test subset of QA-pairs which do not have answer overlap with the training set, which we call the *no overlap test subset*. We further note that we expect higher frequency answers, such as countries, integers and public figures would naturally be expected to appear less often in this test subset. As such, models that perform well on the head of the answer distribution may struggle to perform well in this setting, despite being able to perform some generalisation at test time.

We shall test a variety of models for the answering behaviours using our test subsets. For published models, we obtain test set predictions directly from the authors.

7.4.1 Open-Book Models

See chapter 2 for an introduction to open-book and retrieve-and-read QA models. We consider the extractive retrieve-and-read ODQA model from Karpukhin et al. (2020), which retrieves documents using DPR, before feeding them into an RC model which extracts spans of text as answers. We also include a RAG model, introduced in chapter 6. Finally we include the state-of-the-art Fusion-in-Decoder (FiD, Izacard and Grave, 2021b), which we briefly discussed at the end of chapter 6. FiD is a pipeline retrieve-and-read model using T5-large (Raffel et al., 2020) which retrieves a set of 100 documents and conditions the decoder on all documents at once. FiD differs from RAG in that it isn't trained end-to-end, conditions on more documents, uses a larger model, and fuses information across documents before documents are fed into decoder, rather than afterwards, as in RAG. We not include FiD results on WQ as the authors did not use it in their original work.

7.4.2 Closed-Book Models

See chapter 2 for an introduction to closed-book QA models. In our analysis, we train a BART-large closed-book QA model, which has 400M parameters, which is trained with questions as input and generates QA-pairs as output. Checkpoints are selected by Exact Match score on a development set. We also include a much more powerful T5-11B model from Roberts et al. (2020), with 11B parameters. We use the T5-11B model which has been pretrained with a special "Salient Span Masking" objective (Guu et al., 2020), designed to improve downstream ODQA performance. At the time these experiments were performed, the publicly available T5-11B model checkpoints were trained on both train and development portions of the training QA-pairs, and thus has seen ∼10% more training data than other models. Thus, the scores for this T5 model will in general be slightly higher than they would otherwise be, and hence why the T5 numbers in this chapter are about 2-4% higher than those reported for T5 in chapter 6. Additionally, As we did not include development data in our overlap analysis, a small amount of unaccounted-for overlap occurs for this model. The figures for T5 here are thus a slight overestimate (likely correct to within 5%). We do not include TriviaQA results for T5 as this model was not publicly available when these experiments were performed.

7.4.3 Nearest-neighbour Models: QA-pair Retrievers

Given that there are high levels of train-test overlap in these datasets, we also experiment with some simple nearest-neighbour models. Here, we simply retrieve a QA-pair from the training set based on question similarity to the test question, and return its answer. We experiment with two QA-pair retrievers in this chapter, one using TF-IDF and the other using the dot product similarity of question embeddings from the DPR retriever. These models cannot generalise to non-overlapping answers, and have limited capacity to answer non-overlapping questions. However, these models are attractive from the perspective of model size and efficiency,

		NaturalQuestions				TriviaQA				WebQuestions			
		Total Score	Question Overlap	Answer-Only Overlap	No Overlap	Total Score	Question Overlap	Answer-Only Overlap	No Overlap	Total Score	Question Overlap	Answer-Only Overlap	No Overlap
Open Book	DPR	41.3	69.4	34.6	19.3	57.9	80.4	59.6	31.6	42.4	74.1	39.8	22.2
	RAG-Seq	44.5	70.7	34.9	24.8	56.8	82.7	54.7	29.2	45.5	81.0	45.8	21.1
	FID	51.4	71.3	48.3	34.5	67.6	87.5	66.9	42.8	-	-	-	-
Closed Book	T5 (11B)	36.6	77.2	22.2	9.4	-	-	-	-	44.7	82.1	44.5	22.0
	BART (0.4B)	26.5	67.6	10.2	0.8	26.7	67.3	16.3	0.8	27.4	71.5	20.7	1.6
Nearest neighbour	Dense	26.7	69.4	7.0	0.0	28.9	81.5	11.2	0.0	26.4	78.8	17.1	0.0
	TF-IDF	22.2	56.8	4.1	0.0	23.5	68.8	5.1	0.0	19.4	63.9	8.7	0.0

Table 7.4: EM scores for several recent models on our dataset splits. “Total score” is the overall Test Set EM Score. “Question Overlap” is the test subset with train-test question overlap, and probes for simple question memorisation. “Answer-only Overlap” is the test subset with train-test answer overlap but *without* question overlap, which probes for answer classification. “No overlap” refers to the test subset with no train-test answer overlap, which probes for QA generalisation

amongst the other useful properties that come from non-parametric methods. We shall revisit the subject of QA-pair retrievers in great detail in chapter 8.

7.4.4 Results

Table 7.4 shows the results of our behavioural test set splits for all models. In the following, we shall unpack the major findings:

Question Memorisation Earlier, we found that $\sim$30% of test set questions overlap with the training set. The “Question overlap” columns in Table 7.4 shows performance on Question memorisation. Comparing this column with the total performance column shows that all models perform significantly higher on memorisable questions. This finding is not surprising, but it is worth highlighting that a significant proportion of overall performance is driven by question memorisation. This effect is most pronounced for closed book models. T5-11B performs especially well for question memorisation on both NQ and WebQuestions. This suggests that its extremely large capacity (11 billion parameters), coupled with more powerful question understanding, may allow it to store, recognise and recall training questions more effectively than other models.

Answer Classification The "Answer overlap only" column in Table 7.4 shows performance on answer classification. Answer classification has a large drop in performance compared to question memorisation, dropping by an average of 45%. Open-book models handle this setting better than closed book models. The BART model in particular struggles here, only managing 10.2% EM on this set.

QA Generalisation The "No overlap" column in Table 7.4 shows performance on QA generalisation. All models suffer significant performance degradation on QA generalisation, highlighting the shortcomings of the overall performance metric. For example, we may expect the FiD state-of-the model to answer about half of NQ-style questions correctly, but once we have accounted for repeated questions and answers, it can only answer about one third of questions correctly. This difference is even more pronounced for other models, with an average absolute drop of 25% with respect to overall performance.

Nearest-neighbour Models The bottom two rows of Table 7.4 show the results of our nearest-neighbour QA-pair retrievers. The TF-IDF model, despite being completely untrained, is able to answer about 20% of test questions correctly, purely by retrieving questions from the training sets. More interestingly, the dense retrieval QA-pair retriever outperforms the BART closed-book QA model on NQ and TriviaQA. Furthermore, it also outperforms the significantly more complex DPR open-book model on TriviaQA and WebQuestions on the question overlap subset. These models have severe generalisability limitations, but do represent very space and memory efficient solutions. Our dense nearest neighbour model consists of a single BERT-base checkpoint and outperforms BART-large. The TF-IDF model is even smaller and could be implemented with negligible memory footprint.

7.5 Related Work

In this section, we shall briefly highlight some relevant work that has not already been mentioned in this thesis so far.

Examining what kinds of behaviours are learnt by models has received attention in natural language understanding tasks, such as the GLUE benchmark (Wang et al., 2018a), which includes a diagnostic test set probing for different reasoning types. Lines of work such as Checklist (Ribeiro et al., 2020) extend this to specifying a variety of test-time unit-tests, and assessing how well models pass them.

Various works have also performed critical and careful analysis of question answering systems and datasets. Chen et al. (2016) closely examine the difficulty of the CNN-DM dataset (Hermann et al., 2015), Sugawara et al. (2020) and Kaushik and Lipton (2018) perform an analysis of RC dataset difficulty, and Manjunatha et al. (2019) show that visual QA models memorise common question-answer relationships present in training data. Févry et al. (2020) perform an analysis of various closed-book models' TriviaQA predictions, based on entity mentions. Kwiatkowski et al. (2019) note that the machine reading NQ dataset has substantial train-test overlap of Wikipedia titles, and provide some baselines for "long-answer" QA. Closest to the work in this chapter, Verga et al. (2020) observe similar answer overlap in knowledge-base QA, and explore results on non-overlapping subsets.

7.6 Conclusion

In this chapter, we have, we performed a novel analysis of popular open-domain question answering datasets. We found that 60% of test set answers overlap with the training set and, more surprisingly, 30% of test set questions have at least one duplicate in the train set. Following these observations, we contextualise the performance of seven ODQA models, stratifying by different amounts of training set overlap, gaining an insight into what extent these models generalise or simply memorise their training data. It is clear that performance on these datasets cannot be properly understood by overall QA accuracy and we suggest that in future, a greater emphasis should be placed on more behaviour-driven evaluation, rather than pursuing single-number overall accuracy figures.

We also introduced a novel class of model in the context of modern ODQA, which we refer to as QA-pair retrievers. These models use semi-structured QA-pairs as their units of knowledge representation, rather than parameters or free-text passages. We demonstrated that simple QA-pair retrievers, which work as non-parametric nearest-neighbour models over training QA-pairs, were competitive with closed-book QA models.

In the next chapter, we shall continue our exploration of generalisation vs memorisation in ODQA datasets, and further develop the idea of QA-pairs as a knowledge representation format.

Chapter 8

65 Million Probably-asked Questions and What You Can Do With Them

In chapter 7, we discovered that current closed-book models mostly just memorise training QA-pairs, and can struggle to answer questions that do not overlap with training data. We also proposed a class of simple *non-parametric* knowledge ODQA model, which we referred to as *QA-Pair retrievers*. These explicitly retrieve (training) QA-pairs, rather than memorising them in parameters, and we showed that they performed competitively with closed-book QA models. These models also have a number of useful properties, such as fast inference, interpretable outputs (by inspecting retrieved QA-pairs), and the ability to update the model's knowledge at test time by adding or removing QA-pairs.

However, closed-book QA and QA-pair retriever models are currently not competitive with retrieve-and-read systems in terms of accuracy, largely because the training QA-pairs they operate on cover substantially less knowledge than background corpora like Wikipedia. In this chapter, we explore whether greatly expanding the knowledge coverage of QA-pairs enables closed-book QA and QA-pair retrievers which are competitive with retrieve-and-read models. We shall do this by generating questions from passages in Wikipedia, using a method reminiscent of our first research contribution chapter (chapter 3). However, our aims will be quite different to those from chapter 3 – rather tackling unsupervised RC, we propose to use question generation as a mechanism for transferring knowledge expressed as free text in Wikipedia into the more semi-structured form of question-answer pairs.

Bigger Picture This final chapter focuses on fulfilling the aims that we set out in the introductory chapter that remain unaddressed so far, and bringing together and applying the strands of research we set out so far. Namely, we require models that can store, represent and retrieve knowledge well in order to answer questions with high accuracy, with flexibility, and faster, more efficient inference. We shall use question generation, building on chapter 3, together with lessons learnt about non-parametric memory storage from chapters 5 and 6. We shall use and extend the modelling paradigm introduced from chapter 7, and training algorithms developed from 6. Putting all of these elements together in the right combination, as we shall see, enables us to build models that are significantly stronger than existing systems with respect to our aims. Beyond purely practical performance metrics, we are also continuing to build up our understanding of what behaviours, factors, knowledge and modelling components are required for QA. Accordingly, this chapter provides a deep empirical analysis, comparing different knowledge storage and recall techniques, and a variety of different QA models. Additional commentary on the connections between this chapter and the wider body of work in the thesis can be found in the conclusion of this chapter (sec. 8.7) and the thesis conclusion, chapter 9.

The material in this chapter first appeared in:

> **Patrick Lewis**, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021. PAQ: 65 Million Probably-Asked Questions and What You Can Do With Them. *Transactions of the Association for Computational Linguistics (TACL)*
>
> *Individual Contributions: The original idea of generating data for improved QA-Pair retrievers was conceived by the thesis author. Independently, a similar direction was proposed by a co-author, and the two projects were merged to form a wider collaboration. The study*

design was a close collaboration between all authors. The question-generation pipeline was implemented by co-authors, with the addition of global filtering and scalability factors from the thesis author. The majority of the QA modelling and experiments were performed by the thesis author, with engineering assistance from co-authors. The analysis was a close collaboration between authors, and the majority of the original article was written by the thesis author, with sections being contributed by a number of co-authors.

8.1 Overview

In this chapter, we present Probably-Asked Questions (PAQ), a semi-structured Knowledge Base (KB) of 65M natural language QA-pairs, which models can memorise and/or learn to retrieve from. PAQ differs from traditional KBs in that questions and answers are stored in natural language, and that questions are generated such that they are likely to appear in ODQA datasets. PAQ is automatically constructed using a question generation model applied on Wikipedia. To ensure generated questions are not *only* answerable given the passage they are generated from, we employ a *global filtering* step using an ODQA system. This greatly reduces the amount of wrong/ambiguous questions compared to other approaches (Fang et al., 2020; Alberti et al., 2019), and is critical for high-accuracy downstream QA.

To complement PAQ we develop RePAQ, an ODQA QA-Pair retriever model, using Dense Retrieval (see section 2.3.2.2), and optionally, re-ranking. We show that PAQ and RePAQ provide accurate ODQA predictions, on the level of RAG-Sequence, introduced in chapter 6. PAQ instances are annotated with scores that reflect how likely we expect questions to appear, which can be used to control the memory footprint of RePAQ by pruning the KB accordingly. As a result, RePAQ becomes flexible, allowing us to configure QA systems with near state-of-the-art results, very small memory size, or inference speeds of over 1,000 questions per second.

We also show that PAQ is a useful source of training data for closed-book QA models. BART closed-book QA models trained on PAQ outperform standard data baselines by 5%. However, these models struggle to effectively memorise all the knowledge in PAQ, lagging behind RePAQ by 15%. This demonstrates the effectiveness of RePAQ's non-parametric knowledge architecture for leveraging PAQ.

Finally, we show that as RePAQ's question matching score correlates well with QA accuracy, it effectively "knows when it doesn't know", allowing for *selective question answering* (Voorhees, 2002a) where systems may abstain from answering. Whilst answer abstaining is important in its own right, it also enables an elegant "back-off" approach where we can defer to a more accurate but expensive QA system when the answer confidence is low. This allows us to make use of the best of both speed and accuracy.

In summary, we make the following contributions: i) introduce PAQ, 65M QA-pairs automatically generated from Wikipedia, and demonstrate the importance of global filtering for high quality ii) introduce RePAQ, a QA system designed to utilise PAQ and demonstrate how it can be optimised for memory, speed or accuracy iii) investigate the utility of PAQ for closed-book QA models, improving by 5% but note significant headroom to RePAQ iv) demonstrate RePAQ's strength on selective QA, enabling us to combine RePAQ with a state-of-the-art QA model, making it both more accurate and 2x faster.

8.2 Open-Domain Question Answering

ODQA has been the focus of the majority of this thesis (see section 2.4.2 in the background chapter for a full task description). The goal of ODQA is to develop an answer function $m : Q \mapsto A$, where Q and A respectively are the sets of all possible questions and answers. In this chapter, We assume there is a distribution $P(q,a)$ of QA-pairs, defined over $Q \times A$. A good answer function will minimise the expected error over $P(q,a)$ with respect to some loss function, such as answer string match. In practice, we do not have access to $P(q,a)$, and instead rely on an empirical sample of QA-pairs $\mathcal{K}$ drawn from P, and measure the empirical loss of answer functions on $\mathcal{K}$. Our goal in this chapter is to implicitly model $P(q,a)$ in order to draw a large sample of QA-pairs, PAQ, which we can train on and/or retrieve from.

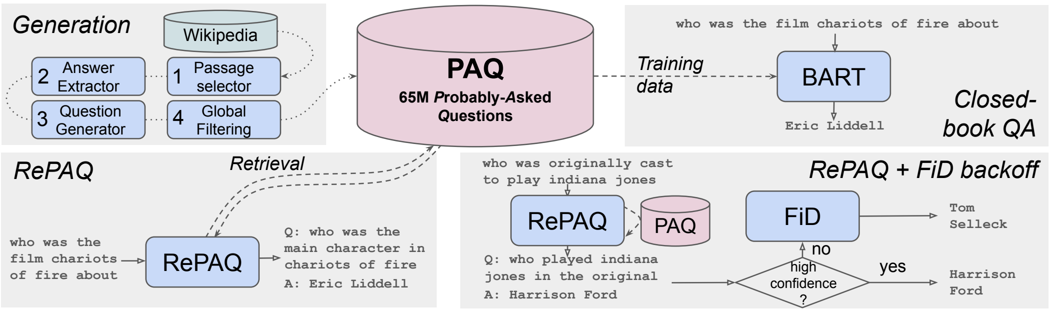


Figure 8.1: Top Left: Generation pipeline for QA-pairs in PAQ. Top Right: PAQ used as training data for closed-book QA models. Bottom Left: RePAQ retrieves similar QA-pairs to input questions from PAQ. Bottom right: RePAQ's confidence is predictive of accuracy. If confidence is low, we can defer to slower, more accurate systems, like FiD.

A sufficiently large drawn sample will overlap with $\mathcal{K}$, essentially *pre-empting* and *caching* questions that humans may ask at test-time. This allows us to shift computation from test-time to train-time compared to retrieve-and-read methods.

8.3 Generating Question-Answer Pairs

In the following, we describe the process for generating PAQ. Given a large textual knowledge source/corpus $\mathscr{C}$, comprised of passages c, our QA-pair generation process consists of the following components:

1. A *passage selection* model $p_s(c|\mathscr{C})$, to identify passages which humans are likely to ask questions about.

2. An *answer extraction* model $p_a(a|c)$, for identifying spans in a passage that are more likely to be answers to a question.

3. A *question generator* $p_q(q|a,c)$ that, given a passage and an answer, generates a question.

4. A *filtering* QA model $p_f(a|q,\mathscr{C})$ that generates an answer for a given question. If an answer generated by p_f does not match the answer a question was generated from, the question is discarded. This ensures generated questions are *consistent* (Alberti et al., 2019).

As shown in Figure 8.1, these models are applied sequentially to generate QA-pairs, similarly to *contextual* QA generation (Alberti et al., 2019; Lewis et al., 2019). First a passage c is selected with a high probability under p_s. Next, candidate answers a are extracted from c using p_a, and questions q are generated for each answer using p_q. Lastly, p_f generates a new answer a' for the question. If source answer a matches a', then (q,a) is deemed consistent and added to PAQ. The pipeline is based on Alberti et al. (2019), updated to take advantage of recent modelling advances. Passage selection and our filtering approach are novel contributions to the best of our knowledge, specifically designed for ODQA QA-pair generation. Each component is described in detail below.

8.3.1 Passage Selection, p_s

The passage selection model p_s is used to find passages which are likely to contain information that humans may ask about, and thus make good candidates to generate questions from. We learn p_s using a similar method to Karpukhin et al. (2020). Concretely, we assume access to a set of positive passages $C^+ \subset \mathscr{C}$, obtained from answer-containing passages from ODQA train sets. As we do not have a set of labelled negatives, we sample negatives either randomly or using heuristics. We then maximise log-likelihood of positive passages relative to negatives. We implement p_s with RoBERTa (Liu et al., 2019c) and obtain positive passages from Natural Questions (NQ). We sample *easy negatives* at random from Wikipedia, and *hard negatives* from the same Wikipedia article as the positive passage. Easy negatives help the model to learn topics of interest, and hard negatives help to differentiate between interesting and non-interesting passages from the same article.

8.3.2 Answer Extraction, p_a

Given a passage, this component identifies spans that are likely to be answers to questions. We consider two alternatives: an off-the-shelf Named Entity Recogniser (NER) or training a BERT answer extraction model on NQ.

The NER answer extractor simply extracts all named entities from a passage.[1] The

[1] We use a spaCy (Honnibal et al., 2019) NER model, trained on OntoNotes (Hovy et al., 2006).

majority of questions in ODQA datasets consist of entity mentions (Kwiatkowski et al., 2019; Joshi et al., 2017), so this approach can achieve high answer coverage. However, as we extract all entity mentions in a passage, we may extract unsuitable mentions, or miss answers that do not conform to the NER system's annotation schema. The trained answer span extractor aims to address these issues.

BERT span extraction is typically performed by modelling answer start and end independently (Devlin et al., 2019). We instead follow the approach of Alberti et al. (2019), which breaks the conditional independence of answer spans by directly predicting $p_a(a|c) = p([a_{\text{st}}, a_{\text{en}}]|c)$. Our implementation first feeds a passage through BERT, before concatenating the start and end token representations of all possible spans of up to length 30, before passing them through an MLP to give $p_a(a|c)$. At generation time, we extract the top-K most probable spans from each passage.

8.3.3 Question Generation, p_q

Given a passage and an answer, this model generates likely questions with that answer. To indicate the answer and its occurrence in the passage, we prepend the answer to the passage and label the answer span with surrounding special tokens. We train on a combination of NQ, TriviaQA, and SQuAD, and perform standard fine-tuning of BART-base (Lewis et al., 2020a) to obtain p_q.

8.3.4 Filtering, p_f

The filtering model p_f improves the quality of generated questions, by ensuring that they are *consistent* – i.e. that the answer they were generated is likely to be a valid answer to the question. Previous work (Alberti et al., 2019; Fang et al., 2020) has employed an RC model for this purpose, $p_f(a|q,c)$, which produces an answer when supplied with a question *and* the passage it was generated from. We refer to this as *local filtering*. However, local filtering will not remove questions which are ambiguous (Min et al., 2020), and can only be answered correctly with access to the source passage. Thus, we use an ODQA model for filtering, $p_f(a|q,\mathscr{C})$, supplied with only the generated question, and *not* the source passage. We refer to this as *global filtering*, and later show it is vital for strong downstream results. We use

FiD-base with 50 passages, trained on NQ (Izacard and Grave, 2021b).

8.4 Question Answering using PAQ

We consider two uses of PAQ for building QA models. The first is to use PAQ as a source of training QA-pairs for closed-book QA models. The second treats PAQ as a KB, which models learn to directly retrieve from. These are related, since we showed in chapter 7 that closed-book QA models mainly just memorise the training QA-pairs into their parameters, latently retrieving from them at test time.

8.4.1 PAQ for Closed-Book QA

We fine-tune BART-large (Lewis et al., 2020a) with QA-pairs from the concatenation of the training data and PAQ, using a similar training procedure to Roberts et al. (2020). We use a batch size of 512, and use validation Exact Match score for early stopping (Rajpurkar et al., 2018). Following recent best practices (Alberti et al., 2019; Yang et al., 2019a), we then fine-tune on training QA-pairs only. We note that effective closed-book QA models must be able to understand the semantics of questions and how to generate answers, in addition to being able to store large numbers of facts in their parameters. This model thus represents a *parametric* knowledgebase and retrieval system. The model proposed in the next section, RePAQ, represents an explicit *non-parametric* instantiation of this idea.

8.4.2 RePAQ

RePAQ is a QA-pair retriever, of the class we introduced in chapter 7. QA-pair retrievers assume access to a KB of N QA-pairs $\mathcal{K} = \{(q_1, a_1), \ldots, (q_N, a_N)\}$. These models provide an answer to a test question q by finding the most relevant QA-pair (q', a') in $\mathcal{K}$, using a scalable relevance function, then returning q' as the answer to q. This function could be implemented using standard information retrieval techniques, (e.g. TF-IDF, like we did in chapter 7) or learnt from training data. RePAQ is a QA-pair retriever which is learnt from ODQA data and consists of a neural retriever, optionally followed by a reranker.

8.4.2.1 RePAQ Retriever

Our retriever adopts the Maximum Inner Product Search (MIPS) dense retrieval paradigm (see section 2.3.2.2 in the background chapter). Our goal is to embed queries q and indexed items d into a representation space via embedding functions g_q and g_d, so that the inner product $g_q(q)^\top g_d(d)$ is maximised for items relevant to q. In our case, queries are questions and indexed items are QA-pairs (q', a'). We make our retriever symmetric by embedding q' rather than (q', a'). As such, *only one* embedding function g_q is required, which maps questions to embeddings. This applies a useful inductive bias, which we find aids stability during training.

Learning the embedding function g_q is complicated by the lack of labelled question pair paraphrases in ODQA datasets. We propose a latent variable approach, very similar to RAG (chapter 6),[2] where we index training QA-pairs rather than documents. For an input question q, the top-K QA-pairs (q', a') are retrieved by a retriever p_{ret} where $p_{\text{ret}}(q|q') \propto \exp(g_q(q)^\top g_q(q'))$. These are then fed into a seq2seq model p_{gen} which generates an answer for each retrieved QA-pair, before a final answer is produced by marginalising,

$$p(a|q) = \sum_{(a',q') \in \text{top-}k \; p_{\text{ret}}(\cdot|q)} p_{\text{gen}}(a|q, q', a') p_{\text{ret}}(q'|q),$$

As p_{gen} generates answers token-by-token, credit can be given for retrieving helpful QA-pairs which do not exactly match the target answer. For example, for the question “when was the last time anyone was on the moon” and target answer “December 1972”, retrieving “when was the last year astronauts landed on the moon” with answer “1972” will help to generate the target answer, despite the answers having different granularity. After training, we discard p_{gen},[3] retaining only the question embedder g. We implement p_{ret} with ALBERT (Lan et al., 2020) with an output dimension of 768, and p_{gen} with BART-large (Lewis et al., 2020a). We train

[2] Other methods, such as heuristically constructing paraphrase pairs assuming that questions with the same answer are paraphrases, and training with sampled negatives would also be valid, but were not competitive in early experiments

[3] We could use p_{gen} as a reranker/aggregator for QA, but in practice find it both slower and less accurate than the reranker described in Section 8.4.2.2

with 100 retrieved QA-pairs, and refresh the index every 5 training steps.

Once the embedder g_q is trained, we build a test-time QA system by embedding and indexing a QA KB such as PAQ. Answering is achieved by retrieving the most similar stored question, and returning its answer. The matched QA-pair can be displayed to the user, providing a mechanism for more interpretable answers than closed-book QA models and many retrieve-and-read generators which consume thousands of tokens to generate an answer. Efficient MIPS libraries such as FAISS (Johnson et al., 2019) enable RePAQ's retriever to answer 100s to 1,000s of questions per second (see Section 8.5.2.3). We use a KB for RePAQ consisting of train set QA-pairs and QA-pairs from PAQ.

8.4.2.2 RePAQ Reranker

Accuracy can be improved using a reranker on the top-K QA-pairs from the retriever. The reranker uses cross-encoding, and includes the retrieved answer in the scoring function for richer featurisation. The model is trained as a multi-class classifier, attempting to classify a QA-pair which answers a question correctly against K-1 retrieved QA-pairs which do not. For each QA-pair candidate, we concatenate the input question q with the QA-pair (q', a'), and feed it through ALBERT, and project the CLS representation to a logit score. The model produces a distribution over the K QA-pairs via softmax, and is trained to minimise the negative log-likelihood of the correct QA-pair.

We obtain training data in the following manner: for a training QA-pair, we retrieve the top $2K$ QA-pairs from PAQ using RePAQ's retriever. If one of the retrieved QA-pairs has the correct answer, we treat it as a positive, and randomly sample K-1 of the remaining retrieved questions as negatives. We train with K=10, and rerank 50 QA-pairs at test time. The reranker improves accuracy at the expense of speed. However, as QA-pairs consist of fewer tokens than passages, the reranker is still faster than retrieve-and-read models, even when using ALBERT-xxlarge.

8.5 Results

We first examine the PAQ resource in general, before exploring how both closed-book QA models and RePAQ perform using PAQ, comparing to recently published systems. We measure performance using Natural Questions (NQ) and TriviaQA, evaluating using standard Exact Match (EM) score. Both datasets, and the evaluation metrics are discussed in detail in chapter 2.

8.5.1 Examining PAQ

We generate PAQ by applying the pipeline described in Section 8.3 to the Wikipedia dump described in section 2.5. We use the passage selection model p_s to rank all passages, and then generate QA-pairs from the top 10M passages – roughly the top 50% of the dump – before applying global filtering.[4]

We are interested in understanding the effectiveness of different answer extractors, and whether generating more questions per answer span leads to better results. To address these questions, we create three versions of PAQ, described below:

- PAQ$_L$ uses a learnt answer extractor, and a question generator trained on NQ and TQA. We extract 8 answers per passage and use beam size 4 for question generation. In PAQ$_{L,1}$ we only use the top-scoring question from the beam.

- PAQ$_{L,4}$ uses the same pipeline as PAQ$_{L,1}$, except that all we use all four questions from the beam, allowing several questions per answer span.

- PAQ$_{NE,1}$ uses the NER answer extractor, and a generator trained on NQ. PAQ$_{NE,1}$ allow us to assess whether diversity in the form of answer extractors and question generators gives better results.

The final KB, referred to as just "PAQ", is the union of PAQ$_L$ and PAQ$_{NE}$.

As shown in Table 8.1, PAQ consists of 65M filtered QA pairs.[5] This was obtained by extracting 165M answer spans and generating 279M unique questions before

[4] Generation was stopped when downstream accuracy with RePAQ did not significantly improve.

[5] Each question only has one answer due to global filtering

Dataset	Extracted Answers	Unique Questions	Filtered QA-pairs	Ratio	Coverage NQ	Coverage TQA
$\text{PAQ}_{L,1}$	76.4M	58.0M	14.1M	24.4%	88.3	90.2
$\text{PAQ}_{L,4}$	76.4M	225.2M	53.8M	23.9%	89.8	90.9
$\text{PAQ}_{NE,1}$	122.2M	65.4M	12.0M	18.6%	83.5	88.3
PAQ	165.7M	279.2M	64.9M	23%	90.2	91.1

Table 8.1: PAQ dataset statistics and ODQA dataset answer coverage. "Ratio" refers to the number of generated questions which pass the global consistency filter.

#	Question	Answer	Comment
1	who created the dutch comic strip panda	Martin Toonder	✓
2	what was the jazz group formed by john hammond in 1935	Goodman Trio	✓
3	astrakhan is russia's main market for what commodity	fish	✓
4	what material were aramaic documents rendered on	leather	✓
5	when did the giant panda chi chi died	22 July 1972	✓, Grammar error
6	pinewood is a village in which country	England	∼, Also a Pinewood village in USA
7	who was the mughal emperor at the battle of lahore	Ahmad Shah Bahadur	✗ Confuses with Ahmad Shah Abdali
8	how many jersey does mitch richmond have in the nba	2	✗ His Jersey No. was 2

Table 8.2: Representative Examples from PAQ. ✓indicates correct, ∼ ambiguous and ✗ incorrect facts respectively

applying global filtering. Table 8.1 shows that the PAQ_L pipeline is more efficient than PAQ_{NE}, with 24.4% of QA-pairs surviving filtering, compared to 18.6%.

PAQ Answer Coverage To evaluate answer extractors, we calculate how many answers in the validation sets of TriviaQA and NQ also occur in PAQ's filtered QA-pairs. Table 8.1 shows that the answer coverage of PAQ is very high – over 90% for both TriviaQA and NQ. Comparing PAQ_L with PAQ_{NE} shows that the learnt extractor achieves higher coverage, but the union of the two leads to the highest coverage overall. Comparing $\text{PAQ}_{L,1}$ and $\text{PAQ}_{L,4}$ indicates that using more questions from the beam also results in higher coverage.

PAQ Question Generation Quality Illustrative examples from PAQ can be seen in Table 8.2. Manual inspection of 50 questions from PAQ reveals that 82% of ques-

#	Model Type	Model	NQ	TQA
1	Closed-book	T5-11B-SSM (Roberts et al., 2020)	34.8	51.0
2	Closed-book	BART-large (Lewis et al., 2021)	26.5	26.7
3	QA-pair retriever	Dense QA-pair Retriever (Chapter 7)	26.7	28.9
4	Open-book, retrieve&read	RAG-Sequence (Chapter 6)	44.5	56.8
5	Open-book, retrieve&read	FiD-large, 100 docs (Izacard and Grave, 2021b)	51.4	**67.6**
6	Open-book, Phrase Index	DensePhrases (Lee et al., 2021)	40.9	50.7
7	Closed-book	BART-large, pre-finetuned on PAQ	32.7	33.2
8	QA-pair retriever	RePAQ (retriever only)	41.2	38.8
9	QA-pair retriever	RePAQ (with reranker)	<u>47.7</u>	50.7
10	QA-pair retriever	RePAQ-multitask (retriever only)	41.7	41.3
11	QA-pair retriever	RePAQ-multitask (with reranker)	47.6	<u>52.1</u>
12	QA-pair retriever	RePAQ-multitask w/ FiD-Large Backoff	**52.3**	67.3

Table 8.3: Exact Match score for highest accuracy RePAQ configurations in comparison to recent state-of-the-art systems. Rows 1-6 are existing systems, Rows 7-12 are models we introduce. Highest score indicated in bold, highest non-retrieve-and-read model underlined.

tions accurately capture information from the passage and contain sufficient details to locate the answer. 16% confuse the semantics of certain answer types, either by conflating similar entities in the passage or by misinterpreting rare phrases (see examples 7 and 8 in Table 8.2). Finally, we find small numbers of grammatical errors (e.g. example 5) and mismatched wh-words (5% and 2% respectively).[6]

Other observations PAQ often contains several paraphrases of the same QA-pair. This redundancy reflects how information is distributed in Wikipedia, with facts often mentioned on several different pages. Generating several questions per answer span also increases redundancy. Whilst this means that PAQ could be more information-dense if a de-duplication step was applied, we later show that RePAQ always improves with more questions (Section 8.5.2.1). This suggests that it is worth increasing redundancy for greater coverage.

8.5.2 Question Answering Results

In this section, we shall compare how the PAQ-leveraging models proposed in Section 8.4 compare to existing approaches. We primarily compare to a state-of-the-art retrieve-and-read model, Fusion-in-Decoder (FiD, Izacard and Grave, 2021b), which was described at the end of chapter 6 and analysed in chapter 7.

[6] Further details in Appendix H.2

#	KB	Filtering Method	Size	NQ Exact Match Score	
				Retriever-only	+ Reranker
1	NQ-Train	-	87.9K	27.9	31.8
2	PAQ$_{L,1}$	None	58.0M	21.6	30.6
3	PAQ$_{L,1}$	Local	31.7M	28.3	34.9
4	PAQ$_{L,1}$	Global	14.1M	38.6	44.3
5	PAQ$_{L,4}$	Global	53.8M	40.3	45.2
6	PAQ$_{NE,1}$	Global	12.0M	37.3	42.6
7	PAQ	Global	64.9M	**41.6**	**46.4**

Table 8.4: The effect of different PAQ subsets on the NQ validation accuracy of RePAQ

Table 8.3 shows the highest-accuracy configurations of our models alongside recent state-of-the-art models. We make the following observations: Comparing rows 2 and 7 shows that a closed-book QA BART model trained with PAQ outperforms a comparable NQ-only model by 5%, and is only 2% behind T5-11B (row 1) which has 27x more parameters. Second, we note strong results for RePAQ on NQ (row 9), actually outperforming RAG-Sequence by 3% (row 4), despite RAG being a retrieve-and-read model.

Multi-task training RePAQ on NQ and TriviaQA improves TriviaQA results by 1-2% (comparing rows 8-9 with 10-11). RePAQ does not perform as strongly on TriviaQA (see Section 8.5.2.6), but is within 5% of RAG, and outperforms concurrent work on real-time QA, DensePhrases (row 6, Lee et al., 2021). Lastly, row 12 shows that combining RePAQ and FiD-large into a combined system is 0.9% more accurate than FiD-large (see Section 8.5.2.4 for more details).

8.5.2.1 Ablating PAQ using RePAQ

Table 8.4 shows RePAQ's accuracy using different PAQ variants. To establish the effects of filtering, we evaluate RePAQ with unfiltered, locally-filtered and globally-filtered QA-pairs from PAQ$_{L,1}$. Rows 2-4 shows that global filtering is crucial, leading to a 9% and 14% absolute improvement over locally-filtered and unfiltered QA-pairs respectively.

We also note a general trend in Table 8.4 that adding more globally-filtered questions improves accuracy. Rows 4-5 show that using four questions per answer span is better than generating one (+0.9%), and rows 5-7 show that combining PAQ$_{NE}$ and PAQ$_{L}$ also improves accuracy (+1.2%). Empirically we did not observe any cases where increasing the number of globally filtered QA-pairs reduced accuracy, even when there were millions of QA-pairs already.

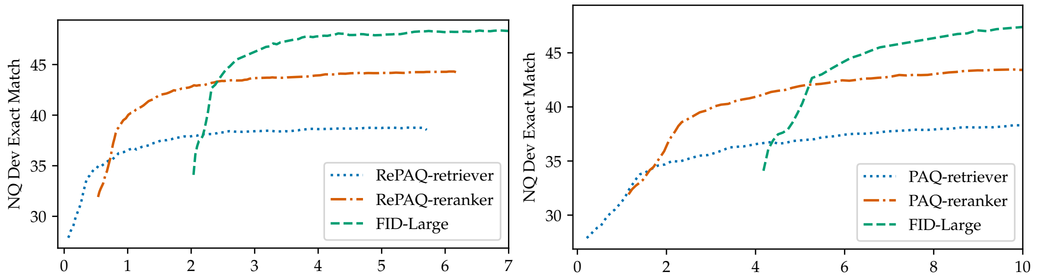
8.5.2.2 System Size vs Accuracy

PAQ's QA-pairs are accompanied by scores of how likely they are to be asked. These scores can be used to filter the KB and reduce the RePAQ system size. A similar procedure can be used to filter the background text knowledge source corpus for a retrieve-and-read model. We compare the system size vs accuracy of a FiD-large system and RePAQ as the number of items (passages and QA-pairs respectively) in their indexes are reduced. We select which passages and QA-pairs are included using the passage selection model p_s. We measure the bytes required to store the models, the text of the documents/QA-pairs, and a dense index.

Comparing systems by size can be nuanced, and sensitive to implementation choices. We take care to minimise these issues by evaluating using two different experimental settings. In Figure 8.2a, We assume models are stored at FP16 precision, the text has been compressed using LZMA[7], and the indexes use 768 dimensional vectors, and Product Quantization (Jégou et al., 2011). These are typical settings when building efficient systems (Izacard et al., 2020; Min et al., 2021). The RePAQ model consists of an ALBERT-base retriever and ALBERT-xxlarge reranker, and the FiD system consists of DPR (Karpukhin et al., 2020) (two BERT-base models) and a T5-large reader (Raffel et al., 2020). Using a different setup (full precision models, no text compression, and FP16 index quantization), is shown in Figure 8.2b. This shifts the position of the curves, but the qualitative relationship is unchanged.

The figures show that both FiD and RePAQ system sizes can be reduced several-fold with only a small drop in accuracy, demonstrating the effectiveness of p_s. FiD can achieve a higher accuracy, but requires larger system sizes. RePAQ can be reduced

[7] https://tukaani.org/xz/



(a) FP16 precision models, LZMA text compression, PQ index compression **(b)** Full Precision models, no text compression, FP16 index compression

Figure 8.2: System Size in GB vs. EM score for RePAQ and FiD-large as a function of the number of items in the index.

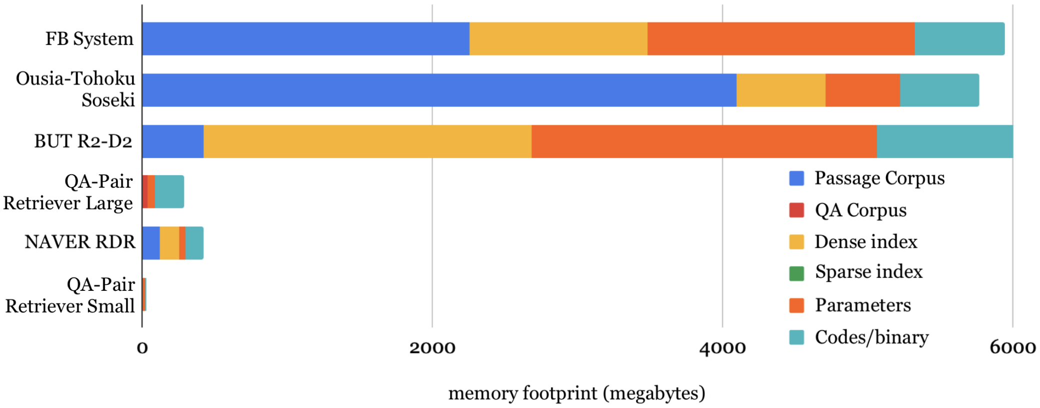


Figure 8.3: Model sizes for winning systems at EfficientQA, reproduced from Min et al. (2021). The QA-pair retrievers are our early prototypes of RePAQ, which won two tracks of the competition.

to a smaller size before a significant accuracy drop, driven primarily by the higher information density of QA-pairs relative to passages, and fewer model parameters used by RePAQ compared to FiD.

We demonstrated the efficacy of QA-pair retrievers at the efficientQA NeurIPS competition (Min et al., 2021). Here, the challenge was to build the smallest ODQA systems, measured by docker image size, while maximising performance on a held-out test set of NaturalQuestions. Our QA-pair retriever entries, which were prototypes of RePAQ, won two tracks of the competition: i) the smallest model capable of 25% accuracy, which we won with a submission of a 29 Mb system scoring 27% EM, and ii) the highest accuracy model less than 500Mb, which we won with a submission of 336Mb which score 33.4% EM. Figure 8.3 shows our system sizes in context with other winners and runners-up in the competition.

Model	Retriever	Reranker	Exact Match	Q/sec
FiD-large	-	-	51.4	0.5
FiD-base	-	-	48.2	2
RePAQ	base	-	40.9	1400
RePAQ	xlarge	-	41.5	800
RePAQ	base	base	45.7	55
RePAQ	xlarge	xxlarge	47.6	6

Table 8.5: Inference speeds of various configurations of RePAQ and FiD on NQ

8.5.2.3 Inference Speed vs Accuracy

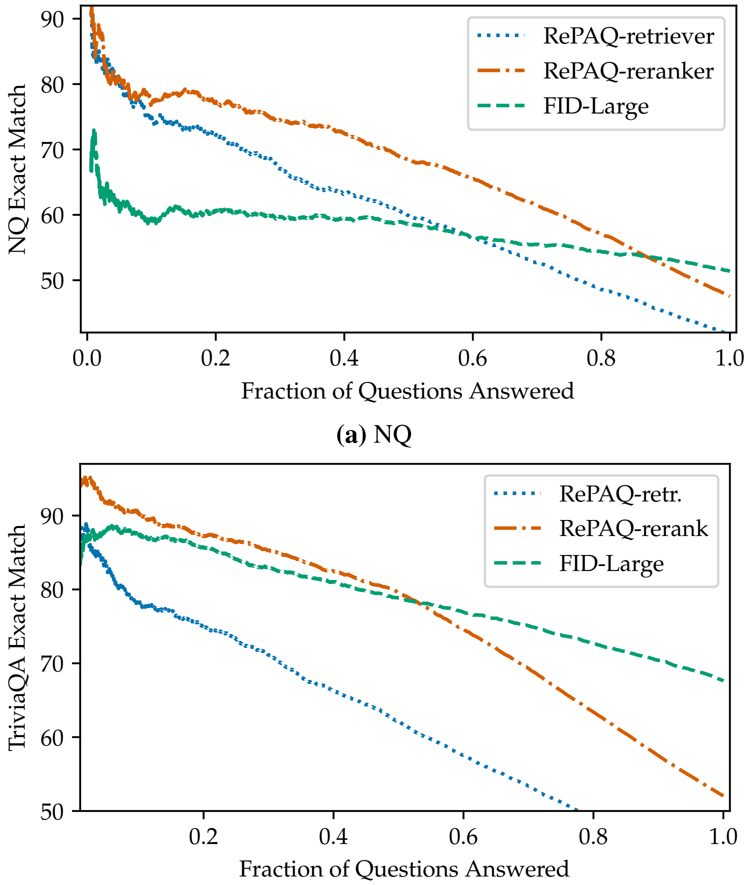
We train a variety of differently sized RePAQ models to explore the relationship between accuracy and inference speed. We use a Hierarchical Navigable Small World (HNSW) index in FAISS (Malkov and Yashunin, 2020; Johnson et al., 2019)[8] and measure the time required to evaluate the NQ test set on a system with access to one GPU (see Appendix H.3 for system details) Table 8.5 shows the results. Some retriever-only RePAQ models can answer over 1,000 questions per second, and are relatively insensitive to model size, with ALBERT-base only scoring 0.5% lower than ALBERT-xlarge. They also outperform retrieve-and-read models REALM (40.4%, Guu et al., 2020) and recent real-time QA models like DensePhrases (40.9%, Lee et al., 2021). We find that larger, slower RePAQ rerankers achieve higher accuracy. However, even the slowest RePAQ is 3x faster than FiD-base, whilst only being 0.8% less accurate, and 12x faster than FiD-large.

8.5.2.4 Selective Question Answering

Models should not just be able to answer accurately, but also "know when they don't know", and abstain when they are unlikely to give good answers (Voorhees, 2002a). This task challenges current models (Asai and Choi, 2021; Jiang et al., 2020b), and has been approached in RC by training on unanswerable questions (Rajpurkar et al., 2018) and using incremental QA for Quizbowl (Rodriguez et al., 2019).

We find that RePAQ's retrieval and reranking scores are well-correlated with answering correctly. RePAQ can thus be used for selective QA by abstaining when the score is below a certain threshold. Figure 8.4a shows a *risk-coverage* plot (Wang

[8] The HNSW index has negligible (∼0.1%) drop in retriever accuracy compared to a flat index



(a) NQ

(b) TriviaQA

Figure 8.4: Risk-coverage plot for FiD and RePAQ for a) NQ and b) TriviaQA

et al., 2018e) for RePAQ and FiD on NQ, where we use FiD's answer log probability for its answer confidence.[9] The plot shows the accuracy on the top N% highest confidence answers for NQ. If we require models to answer 75% of user questions, RePAQ's accuracy on the questions it does answer is 59%, whereas FiD, which has poorer calibration, scores 55%. This difference is more pronounced with stricter thresholds: at 50% coverage, RePAQ outperforms FiD by over 10%. FiD only outperforms RePAQ when we require systems to answer over 85% of questions.

For TriviaQA (Figure 8.4b), the results are qualitatively similar to NQ, although FiD's stronger overall performance shifts its risk-coverage curve up the accuracy axis relative to RePAQ. FiD also appears to be better calibrated on TriviaQA than it

[9] We also investigate improving FiD's calibration using an auxiliary model, see Appendix H.4. We find that the most effective way to calibrate FiD is to use RePAQ's confidences

Input: who was the film chariots of fire about	**A**: Eric Liddell	
who was the main character in chariots of fire	**A**: *Eric Liddell*	✓
who starred in the movie chariots of fire	**A**: Ian Charleson	✗
which part did straan rodger play in chariots of fire	**A**: Sandy McGrath	✗
who played harold in the 1981 film chariots of fire	**A**: Ben Cross	✗
who is the main character in chariots of fire	**A**: Eric Liddell	✓
Input: what is the meaning of the name didymus	**A**: twin	
what language does the name didymus come from	**A**: Greek	✗
where does the name didymus come from in english	**A**: Greek	✗
what does the word domus mean in english	**A**: home	✗
how long has the term domus been used	**A**: 1000s of years	✗
what does the greek word didyma mean	**A**: *twin*	✓
Input: what is the name of a group of llamas	**A**: herd	
what are llamas and alpacas considered to be	**A**: domesticated	✗
what are the figures of llamas in azapa valley	**A**: Atoca	✗
what are the names of the llamas in azapa valley	**A**: Atoca	✗
what is the scientific name for camels and llamas	**A**: *Camelidae*	✗
are llamas bigger or smaller than current forms	**A**:larger	✗

Table 8.6: Examples of top 5 retrieved QA-pairs for NQ. Italics indicate QA-pairs chosen by reranker.

is for NQ, indicated by higher gradient. However, RePAQ remains better calibrated, outperforming it for answer coverages below 50%.

Whilst RePAQ's selective QA is useful in its own right, it also allows us to combine the slow but accurate FiD with the fast and precise RePAQ, which we refer to as *backoff*. We first try to answer with RePAQ, and if the confidence is below a threshold determined on development data, we pass the question onto FiD. For NQ, the combined system is 2.1x faster than FiD-large, with RePAQ answering 57% of the questions, and the overall accuracy is 1% higher than FiD-large (Table 8.3).

If inference speed is a priority, the threshold can be decreased so that RePAQ answers 80% of the questions, which retains the same overall accuracy as FiD, with a 4.6x speedup. For TriviaQA, the combined system backs off to FiD earlier, due to the stronger relative performance of FiD.

8.5.2.5 Analysing RePAQ's Predictions

Some examples of top retrieved questions are shown in Table 8.6. When RePAQ answers correctly, the retrieved question is a paraphrase of the test question from

PAQ in 89% of cases. As such, there is high (80.8 ROUGE-L) similarity between correctly answered test questions and the top retrieved questions. 9% of test questions even exist verbatim in PAQ, and are thus trivial to answer. The reranker primarily improves over the retriever for ambiguous cases, and cases where the top retrieved answer does not have the right granularity. In 32% of cases, RePAQ does not retrieve the correct answer in the top 50 QA-pairs, suggesting a lack of coverage may be a significant source of error. In these cases, retrieved questions are much less similar to the test question than for correctly answered questions, dropping by 20 ROUGE-L. We also observe cases where retrieved questions match the test question, but the answer does not match the desired answer. This is usually due to different answer granularity, but in a small number of cases is due to factually incorrect answers.

8.5.2.6 Does the Filtering Model Limit RePAQ's Accuracy?

As RePAQ relies on retrieving paraphrases of test questions, we may expect that the ODQA filtering model places an upper bound on it's performance. For example, if a QA-pair is generated which overlaps with a test QA-pair, but the filter cannot answer it correctly, that QA-pair will not be added to PAQ, and RePAQ cannot use it to answer the test question. The NQ FiD-base-50-doc model used for filtering scores 46.1% and 53.1% for NQ and TriviaQA respectively. RePAQ actually outperforms the filter model on NQ by 1.6%. This is possible because generated questions can be phrased in such a way that they are easier to answer, e.g. being less ambiguous (Min et al., 2020). RePAQ *can* then retrieve the paraphrased QA-pair and answer correctly, even if the filter could not answer the test question directly. The filtering model's weaker scores on TriviaQA helps explain why RePAQ is not as strong on this dataset. We speculate that a stronger filtering model for TriviaQA would in turn improve RePAQ's results.

8.5.3 Closed-book QA vs RePAQ

Table 8.7 shows results on the behavioural memorisation test set splits which we introduced in chapter 7. These splits measure how effectively models memorise QA-

Model	Training QA-Pairs / QA-Pairs KB	Total Score	Question Overlap	Answer-Only Overlap	No Overlap
Closed-book QA BART	NQ	26.5	67.6	10.2	0.8
Closed-book QA BART	NQ+PAQ	28.2	52.8	24.4	9.4
+ final NQ finetune	NQ+PAQ	32.7	69.8	22.2	7.5
RePAQ	NQ	31.3	78.1	14.3	0
RePAQ	NQ+PAQ	47.3	73.5	39.7	26.0

Table 8.7: Results for closed-book BART and RePAQ on the QA-overlap behavioural splits from chapter 7 for NQ.

pairs from the NQ train set (“Question overlap”), and generalise to novel questions (“Answer-only overlap” and “No overlap”). Comparing closed-book QA models trained on NQ vs those trained on NQ and PAQ show that models trained with PAQ answer more questions correctly from the “Answer-only overlap” and “No overlap” categories, indicating that they have learnt facts *not* present in the NQ train set. Applying further NQ finetuning on the PAQ closed-book QA model improves scores on “Question overlap” (indicating greater memorisation of NQ), but scores on the other categories drop (indicating reduced memorisation of PAQ).

RePAQ, which explicitly retrieves from PAQ rather than trying to memorise it in parameters, strongly outperforms the closed-book QA model in all categories, demonstrating that the closed-book QA model struggles to memorise enough facts from PAQ. Note how when RePAQ is supplied with NQ, its performance on “No-overlap” questions is zero, i.e. it cannot generalise to answers not present in its training data. However, when it is supplied with PAQ, it scores 26% in this generalisation category, effectively achieving generalisation by “memorisation” of PAQ.

We note that higher parameter-count closed-book QA models be better able to memorise PAQ, but have downsides in terms of system resources. Future work should address how to better store PAQ in closed-book QA model parameters.

8.6 Related Work

In this section, we shall highlight work of specific relevance to this chapter, which has not already been discussed.

KBQA A number of early approaches in ODQA focused on using structured KBs (Berant et al., 2013) such as Freebase (Bollacker et al., 2008), with recent examples from Févry et al. (2020) and Verga et al. (2020). This approach often has high precision but suffers when the KB doesn't match user requirements, or when the schema limits what knowledge can be stored. We populate our KB with semi-structured QA-pairs, specifically designed to be relevant at test time, mitigating such drawbacks, but sharing benefits such as precision and extensibility.

OpenIE Our work in this chapter touches on KB construction and open information extraction (OpenIE) (Angeli et al., 2015). Here, the goal is to extract structured or semi-structured facts from text, typically (subject, relation, object) triples for use in tasks such as slot-filling (Surdeanu, 2013). We generate natural language QA-pairs rather than OpenIE triples, and do not attempt to extract all possible facts in a corpus, focusing only on those likely to be asked. QA-pairs have also been used in semantic role labelling, e.g. QA-SRL (FitzGerald et al., 2018).

Predictive Annotation, Phrase Indices and Real-time ODQA (Prager et al., 2000) introduce an early phrase indexing method, where potential answers in text are identified, and indexed. This method resembles modern phrase index models, and the answer extraction part of the PAQ generation pipeline. Systems prioritising fast runtime over accuracy are sometimes referred to as *real-time QA* systems (Seo et al., 2018). DenSPI (Seo et al., 2019) and followup work, DensePhrases (Lee et al., 2021), index all possible phrases in a corpus, and learn mappings from questions to passage-phrase pairs. These methods are described in more detail in section 2.4.2.3 in the background chapter. We also build an index for faster answering, but generate and index globally answerable questions rather than phrases. Indexing QA-pairs could be considered as indexing summaries of important facts from the corpus, rather than indexing the corpus itself. We also generate and store *multiple* questions per passage-answer pair, relieving information bottlenecks from encoding a passage-answer pair into a single vector.

Question Generation for QA Question generation has been used for various purposes, such as data augmentation (Alberti et al., 2019; Lee et al., 2021), improved retrieval (Nogueira et al., 2019), generative modelling for contextual QA (Lewis and Fan, 2018), as well as being studied in its own right (Du et al., 2017; Hosking and Riedel, 2019). We also used it to induce unsupervised RC models in chapter 3. Serban et al. (2016) generate large numbers of questions from Freebase, but do not address how to use them for QA. Closest to our work is the recently proposed OceanQA (Fang et al., 2020). OceanQA first generates contextual QA-pairs from Wikipedia. At test-time, a document retrieval system is used to retrieve the most relevant passage for a question and the closest pre-generated QA-pair from that passage is selected. In contrast, we focusing on generating a large KB of non-contextual, globally consistent ODQA questions and explore what QA systems are facilitated by such a resource.

8.7 Discussion and Conclusion

In this chapter, we have introduced a dataset of 65M QA-pairs, and explored its uses for improving non-traditional ODQA models. We demonstrated the effectiveness of RePAQ, a QA-pair retriever class model which retrieves from PAQ, in terms of accuracy, speed, space efficiency and selective QA.

We also demonstrated PAQ's utility for improved closed-book QA, but note a large accuracy gap between our closed-book QA models and RePAQ. Exploring the trade-offs between storing and retrieving knowledge parametrically or non-parametrically is of great current interest, both in our thesis, and in general in the field (Lewis et al., 2020b; De Cao et al., 2021b), and PAQ should be a useful testbed for probing this relationship further. We also note that PAQ could be used as general data-augmentation when training any open-domain QA model or retriever. We consider such work out-of-scope here, but we note that Oguz et al. (2021) have reported using PAQ as data augmentation improves DPR on NQ by 9% recall@5, and R-precision on the KILT benchmark by 7% (Petroni et al., 2021). This retrieval improvement will, in turn, transfer to stronger downstream retrieve-and-read models.

We found that RePAQ's errors are driven by a lack of coverage, thus generating more QA-pairs could improve accuracy further. However, phenomena such as compositionality will eventually impose practical limits on this approach. Multi-hop RePAQ extensions suggest themselves as ways forward here, as well as back-off systems (see Section 8.5.2.4). Indeed, models like RePAQ will be best utilised in back-off configurations in practical settings, where it may be important to capture the tail of the distribution and generalise, something that the latest retrieve-and-read models like FiD are stronger for, but also to exploit the benefits of speed and controlability in the average case, which RePAQ is well suited for.

A key limitation is that generating PAQ-style collections is computationally intensive due to its large scale and global filtering requirements. This being said, it should be a useful, re-usable resource for researchers. Since its public release, PAQ has been used for a variety of purposes beyond those we use it for in this chapter (Oguz et al., 2021; Sciavolino et al., 2021; Kharitonov et al., 2021; Ye et al., 2021; Chen et al., 2021). Nevertheless, future work should be carried out to improve the efficiency of generation, in order to improve applicability. Relaxing the requirement for global filtering should be a key target for future work, since despite its efficacy, it is the most expensive step by far, and also aggressively filters generated questions, placing an upper bound on coverage, and hence the downstream performance.

Chapter 9

Conclusions

In this thesis, we have explored a series of problems in text-based RC and ODQA by employing retrieval and generation techniques. Following introductory and background material in Chapters 1 and 2, we have considered how to perform RC without RC annotations in Part I, retrieval-augmented ODQA in Part II and explored the relationship between memorisation and generalisation in ODQA in Part III.

9.1 Summary of Contributions

In this section, we shall summarise our major contributions and findings, grouped qualitatively into themes.

9.1.1 Datasets

We have developed and publicly released a number of new datasets and resources for the research community. We have introduced a large, high-quality evaluation dataset for multi-lingual RC in 7 languages, which has seen extensive subsequent adoption both for multi-lingual QA, but also for general multilingual NLU, including integration into aggregated multilingual NLU benchmarks XGLUE (Liang et al., 2020) and XTREME (Hu et al., 2020). We have also provided generalisation meta-annotations for popular ODQA test sets. These annotations enable stronger benchmarking, and have been used widely by the community (Cheng et al., 2021; Fajcik et al., 2021; Reddy et al., 2021b; Mao et al., 2021) Finally, we introduced a very large resource of high-quality generated question-answer pairs, which has subsequently seen use for dense retrieval (Oguz et al., 2021; Sciavolino et al., 2021) and as a tool for studying memorisation (Kharitonov et al., 2021) and distillation (Ye et al., 2021; Chen et al., 2021).

9.1.2 Unsupervised QA and Zero-Shot Transfer

We have introduced the task of unsupervised RC, and proposed a method which enables unsupervised RC at the level of early supervised approaches. Subsequent work has built on our contribution, spawning a sub-field in the RC research community (Fabbri et al., 2020; Li et al., 2020; Hong et al., 2020; Bian et al., 2021). We have also made contributions towards unsupervised ODQA, demonstrating that unsupervised models can outperform early supervised ODQA systems. Finally, we performed a detailed study of zero-shot cross-lingual transfer in RC, and demonstrated that multi-lingual pretrained models show great promise towards QA systems which do not require in-language training data.

9.1.3 Retrieval-Augmented Models

We demonstrated the effectiveness of retrieval-augmentation to improve the factuality of pretrained language models. In addition, we introduced a class of retrieval-augmented, end-to-end-trainable seq2seq model. This model is trained using only pairs of input-output sequences, and acts as a drop-in replacement for popular pretrained seq2seq models. We demonstrated the flexibility of this model across classification, short-answer and generation formats, and showed its strong performance on knowledge-intensive tasks. Partially inspired by our contributions, this area has spawned exciting developments in retrieval-augmented models in a wide range of applications, such as dialogue (Shuster et al., 2021), slot-filling (Glass et al., 2021), code generation (Liu et al., 2021b) and KBQA (Das et al., 2021).

9.1.4 Parametric vs non-Parametric Knowledge

We have investigated the strengths and weaknesses of parametric and non-parametric methods of storing knowledge. We have demonstrated that where parametric knowledge is present, augmentation with non-parametric knowledge improves our ability to leverage it. We have further explored limitations of parametric memory, demonstrating that parametric knowledge models suffer from a capacity problem, whereas non-parametric techniques are not limited in the same way. Whilst this may be mitigated by the development of ever larger parameter-count models, non-parametric knowledge has the advantage of being flexible, and effectively unbounded. Lastly, we have demonstrated the propensity of parametric-memory models to memorise their fine-tuning data, as well as struggling to apply knowledge from pretraining time, with has implications for closed-book QA.

9.1.5 Flexible, Efficient and low-latency ODQA Models

Besides the aforementioned flexible retrieval-augmented seq2seq models, we developed a class of ODQA model based on the concept of retrieving from a semi-structured knowledgebase of question-answer pairs. We demonstrated this approach's low-latency compared to other methods, and validated its space-efficiency in community shared tasks. We also showed its strength in selective QA, and developed a simple but effective strategy for how to combine it with slower, more general models to achieve a combination of low-latency and state-of-the-art accuracy.

9.2 Limitations and Future Work

Finally, we shall revisit our main contribution areas to summarise their limitations, and indicate areas for future work, before moving onto some broader trends that we expect to see in ODQA and knowledge-intensive NLP in upcoming years.

9.2.1 Direct Reflections and Future Work

9.2.1.1 Datasets

There are a number of limitations of MLQA. First, since MLQA is designed to closely match SQuAD v1, it inherits many of its limitations, such as overly-encouraging lexical and answer-type pattern matching. Indeed our dataset construction method may exacerbate such issues, especially if one were to train on instances from MLQA. Moreover, our dataset construction method, and those of contemporary efforts (Artetxe et al., 2020) are at risk of overly-representing topics of interest to English-speaking cultural backgrounds. This could come at the expense of the cultures the languages that we evaluate on are rooted in. We also note that whilst we were able to significantly reduce the amount of manual translation, which is known to cause data artefacts (Lembersky et al., 2011; Volansky et al., 2015), we did not remove it entirely. We highlight TyDiQA (Clark et al., 2020) as a promising step forward in terms of multilingual QA design, which address many of these limitations, despite lacking parallel instances, issues with cross-language comparisons, and still relying on crowdsourced questions. Lastly, we did not annotate unanswerable questions in our dataset, which have been shown to be important for high-quality RC systems (Rajpurkar et al., 2018). We further note that whilst the multilingual RC task is now relatively well-established, the area of multilingual ODQA is some way behind. We see promising datasets in this area from Asai et al. (2021) and Longpre et al. (2021), although these re-purpose existing datasets (TyDiQA and NQ respectively), inheriting some of their issues.

With respect to our behavioural annotations on existing ODQA datasets, we note that the number of instances in our annotated subsets are small, making cross-model comparison less reliable. We also note our annotations only represent a small facet of generalisation behaviour. In particular, whilst our annotations identify questions that only require simple memorisation, they tell us relatively little about the reasoning skills required to answer other types of questions. We believe there is a great value in meta-annotating popular test sets to better understand the generalisation and reasoning abilities of current and future models. We hope to see more area in this work, in the vein of Sciavolino et al. (2021), who recently demonstrated the weakness of dense retrievers on entity-centric questions and Liu et al. (2021a) who develop annotations for compositional generalisation. We hope that future datasets and tasks should be developed with behavioural splits specifically in mind, in order to avoid periods of misleading evaluation.

Lastly, turning to PAQ, we note that whilst our generation method is empirically effective, its major limitation is its computational cost. The global filtering method is important for high-quality generation, but it is slow and inefficient, limiting applicability. Moreover, the global filtering requirement presupposes the availability of a high-quality ODQA system. Even with global filtering, a significant number of incorrect questions pass through into the dataset, and a high number of questions that are correct get removed due to the limited accuracy of the filter. Future work should look to lift the requirement to apply a global filter, and more generally work to improve efficiency for factual, non-hallucinatory question generation.

9.2.1.2 Unsupervised QA and Zero-Shot Transfer

Our work in unsupervised RC has a number of limitations. Firstly, we assume access a high-quality prior over answers, in the form of a noun phrase or entity mention recognition model, in addition to a constituency parser. If these requirements could be lifted, it would greatly open up the applicability of the technique. Furthermore, our unsupervised question generator generates a substantial number of low quality or accidentally unanswerable questions. Moreover, the QA behaviour induced in the final model is basic, and brittle. A number of works have suggested improvements to our technique, with some modest gains in accuracy, but this usually comes with additional heuristics. The unsupervised requirement for RC, whilst an interesting and informative exercise, is perhaps counterproductive for creating practical low data RC models. It is our opinion that few-shot and transfer-learning approaches are more appropriate going forward. This area has greatly increased in popularity recently, and we expect there to be many innovations in few-shot modelling that will directly feed into better low-data RC models.

Our work in zero-shot ODQA also has limitations. We only tackle the task of cloze-style ODQA with single-word answers, drawn from the LAMA probe (Petroni et al., 2019). Answering natural questions, and questions with multi-word answers represents important future work in this area. We would also like to delineate two potentially-confounded reasons to study unsupervised ODQA. The first is for probing how well models capture and apply knowledge. The second is for attempting to build stronger ODQA models, without using annotations. Our work in this area falls into the former category. Future work that tackles the latter should be careful about using datasets that were created for the former, and unintentionally "overfitting". As a result, we suggest such work should evaluate against standard ODQA datasets such as NQ and TriviaQA, rather than probing style ones like LAMA, following the example of recent work from Zhu et al. (2021). We would also advocate that such non-probing work would be more appropriately formulated as a few-shot task. We also demonstrated the utility of BERT's next-sentence-prediction task for zero-shot ODQA, which increased the model's robustness to noise in retrieved documents. Such pretraining objectives have fallen out of favour, likely due to an emphasis on evaluation procedures rewarding other behaviours. We suggest that future work considering pretraining objectives from a wider perspective would be fruitful.

9.2.1.3 Retrieval-Augmented Models

The retrieval-augmented generators presented in chapter 6 can struggle to surface relevant documents for a downstream task that is very different to the one they were pretrained on. The generator is also free to learn to only weakly-condition on retrieved documents, or even ignore them. These two issues combined can lead to instability. Future work to address these key issues is needed in order to truly realise the potential of this modelling framework. This requires improvements to more generalised ad-hoc retrievers, with recent promising developments spurred by benchmarks designed to test the generalisation in retrieval (Thakur et al., 2021). Future work on training objectives which encourage retrieval-augmented generators to make the maximum use of the documents should also improve stability. Models which pass messages between documents earlier, such as FiD (Izacard and Grave, 2021b) empirically outperform RAG, but are challenging to train end-to-end. Future work to train earlier fusion models end-to-end will be fruitful, and we are excited by promising recent attempts, such as by Sachan et al. (2021).

9.2.1.4 Parametric vs Non-Parametric Models

An intense interest in prompting (which the retrieval-augmentation strategy from chapter 5 could be considered an input-dependent special case of) has recently emerged. We expect that this interest, mostly motivated for few-shot learning, will also translate to improved parametric knowledge probing. Indeed, we have already seen this line of work begin to bear fruit, with auto-prompting and continuous prompting work showing strong results on LAMA (Jiang et al., 2020a; Shin et al., 2020; Liu et al., 2021c; Zhong et al., 2021; Qin and Eisner, 2021). Many of these probing advances are orthogonal to retrieval-augmentation, and could be applied in combination in future work. Moreover, optimising probing techniques specifically for retrieval-augmented inputs could be fruitful way forward for strong hybrid knowledge models. Taking a step back, work on systematic description and evaluation of parametric knowledge in needed, as well as additional and stronger probing sets. Efforts like BeliefBank (Kassner et al., 2021) show promise here.

We also highlighted updatability and interpretability issues for parametric knowledge. Future work that attempts to shed light on closed-book QA would aid interpretability, with influence functions perhaps being a promising way forward (Koh and Liang, 2017; Chen et al., 2020). Updating the knowledge in pretrained models is also an area in its infancy (De Cao et al., 2021a; Dai et al., 2021).

Understanding how parametric knowledge scales with parameter count is an active area of research. In our contributions we limited ourselves mostly to models below one billion parameters. However, it is increasingly clear that models beyond one billion parameters can behave qualitatively differently (Brown et al., 2020; Kirstain et al., 2021, inter alia.) As a result, caution should be exercised in directly extrapolating the results we have observe onto extremely large models. Nevertheless, we expect that non-parametric components will still remain an effective way of obtaining knowledge for modelling, whilst being much more energy-efficient and accessible, and thus have an important part to play in future model development.

9.2.1.5 Efficient and low-latency Models

We have demonstrated effective strategies for efficient and low-latency test-time ODQA models. However, it is worth highlighting a few limitations. First, although we are able to achieve space-efficiency, low-latency and accuracy using RePAQ, it is still challenging to find operating points for a single model that perform well on all three criteria simultaneously. A further issue is that whilst we can build very space-efficient and low-latency models at test time, this is not the case for training time. In particular, as mentioned above, the generation process for PAQ, which is required for RePAQ to perform well, is slow and expensive relative to training standard models. This is justifiable if the generated resource sees significant use, which we have seen with PAQ, but in general, is not a sustainable strategy, unless generation can be made significantly faster and cheaper. Trading expensive training for cheaper inference is also justifiable if the model will see significant inference usage, such as being employed in a web-scale product. Making faster, smaller and cheaper ODQA models which can still retain accuracy is an important area for future work. We hope that the popularity of efforts like EfficientQA (Min et al., 2021) will grow, enabling this area to flourish.

9.2.2 Broader Outlook

Here, we draw the thesis to a close by picking up on some broader areas for future work, and some trends we expect, or hope, to see in the field in future.

Beyond factoid and span-based QA We have concerned ourselves mainly with span-based, short answer factoid QA. This is attractive for its comparatively simple modelling paradigm, and its ease of evaluation. However, ultimately, we should move beyond this paradigm, towards fully free-form answers. The length and depth of an answer should be predicated on the type of question being asked, as well as some contextual knowledge about the person or entity asking the question. For example, asking for a description or explanation should elicit a relatively long answer, whereas a simple relational questions should receive short answers. Datasets promoting this area have been proposed, such as Eli5 (Fan et al., 2019) amongst others, but useful, non-hallucinatory systems have yet to firmly establish themselves. We suggest that the fields of knowledge-grounded dialogue and ODQA, which broadly seek to solve the same set of problems, will grow closer and perhaps coalesce for the next generation of QA (Qu et al., 2020; Anantha et al., 2021).

Evaluation In general, evaluation methods in the area of ODQA are in need of improvement. Our current popular evaluation metrics (Exact Match, F1) are relatively crude assessors of QA performance, due to limited numbers of references, and lack answer equivalence recognition. Careful development of model-assisted evaluation metrics may be appropriate. The need for stronger evaluation protocols is more pressing if we are to expand more confidently into free-form answering, where standard EM and F1 break down, and metrics such as BLEU and ROUGE are not significantly better. We also need to establish and adopt procedures for assessing equity, societal bias and other harms in QA models. Whilst this is a problem across NLP, it is perhaps more pressing in QA systems, which are more likely to be user-facing than most models, and thus more capable of harm. This must be assessed from the perspectives of how models behave on certain types of questions, but also by examining the appropriateness and representative nature of the knowledge sources we use as well. Recent work suggest that whilst current ODQA models may not exhibit acutely problematic behaviour on standard benchmarks, the benchmarks themselves do not contain a broad-enough coverage topics to properly assess demographic issues (Gor et al., 2021).

Another broad key area for future work is the proper assessment of distributional change for ODQA systems. ODQA systems generalise relatively weakly across different test question datasets, and can even degrade on datasets collected using identical methodology, collected a few weeks apart (Min et al., 2021). It is highly likely that state-of-the-art ODQA models generalise poorly across time, and their performance on time-static baselines is misleading. Thus, establishing how quickly performance degrades with time, and developing an agreed-upon metric for measuring lifelong performance is important (Lazaridou et al., 2021). This will require us to reduce our reliance on static test sets (Kiela et al., 2021). In turn this should encourage the development of life-long learning models, an area that deserves great attention, especially as pretraining becomes more expensive.

Growth of parametric knowledge It is inevitable that model parameter counts will continue to grow, and this will be accompanied by enhanced parametric knowledge. As such, we expect very large parametric knowledge-only models to soon perform competitively with other ODQA approaches. We reiterate that this comes with drawbacks in terms of interpretability, providence, updatability and efficiency. There is much work to do in order to mitigate these issues before closed-book QA models can be used responsibly in practical QA applications.

Multi-modality There are already deeply established lines of work on QA with textual knowledge sources, and structured KBQA, and QA leveraging semi-structured sources. Work on models that draw on several modalities to answer questions is less common, but is growing in popularity (Talmor et al., 2021; Oguz et al., 2020). Knowledge that is frequently expressed in one modality may be less prevalent in another, such as the action of physics on objects being much more obvious in video than text. Thus tackling multi-modal knowledge is important for building complete-knowledge models, which we expect to be a prominent theme in future work.

Green AI There is growing awareness of the environmental impact of AI research (Strubell et al., 2019). We expect work that emphases more efficient models to be high-impact, and hardware-software co-design for neural models to become increasingly relevant. Tools to track the carbon emissions from research should be widely adopted, and estimated emissions reported in publications (Lacoste et al., 2019; Schmidt et al., 2021). The environmental impacts of this thesis are assessed in Appendix B.2.

Appendix A

Open-sourced Materials

- Code, models and data to support chapter 3 can be found at https://github.com/facebookresearch/UnsupervisedQA
- The MLQA dataset and evaluation scripts introduced in chapter 4 are publicly available at https://github.com/facebookresearch/mlqa
- The LAMA dataset and tools for running experiments such as those in chapter 5 are available at https://github.com/facebookresearch/LAMA
- RAG is freely-available as part of HuggingFace Transformers (Wolf et al., 2020) at https://github.com/huggingface/transformers. Scripts to run experiments with RAG can be found at https://github.com/huggingface/transformers/blob/master/examples/rag/README.md and an interactive demo of a RAG model can be found at https://huggingface.co/rag/
- Data and evaluation code to support the analysis in Chapter 7 is publicly available at https://github.com/facebookresearch/QA-Overlap
- Data, models and code supporting chapter 8 are available at https://github.com/facebookresearch/PAQ. This includes all the PAQ data and RePAQ models, including a memory-efficient RePAQ retriever designed for use with modest hardware.

Appendix B

Engineering Details

B.1 Hardware Details

The majority of experiments performed in this thesis were performed using a machine learning workstation with 80 CPU cores, 512GB of CPU RAM and access to 8 32GB NVIDIA V100 GPUs. For the experiments in chapters 3, 4, 5, only one GPU was used for training and inference. For chapter 6, models were trained with either 8 GPUs using the whole workstation node, or occasionally 16 GPUs for expedited training, using two nodes, and up to 128 GB of CPU RAM was required for developing indexes. For chapter 7, experiments did not require any significant computational resources. Lastly, for chapter 8, which uses the heaviest use of compute, 8 GPUs were used for training models, and up to 128 GB of CPU RAM was required for indexes. See Appendix H.5 for further details on training for the RePAQ and Closed-book QA models in chapter 8. Generation of PAQ was parallelised across 100 GPUs (distributed over a compute cluster) for approximately 1 week of continuous runtime.

Hyperparameter sweeps were used throughout the experiments when training models in this thesis, which were usually parallelised across a compute cluster to save time, with each parallel job requiring the resources stated above. Typically these sweeps would be over a handful (∼2-10) of hyperparameter settings. Inference for all models developed in this thesis can be run on a single GPU, and require up to 32GB of CPU memory for indices.

B.2 Environmental Impact

All experiments were carried out using electricity from 100% renewable, non-CO_2 emitting sources, in a data-centre with a power-effective usage (PUE) index of 1.1. In theory, the experiments run in this thesis produced net-zero CO_2 emissions. However, the picture is complex, and the emissions embodied in the construction of the hardware, and other factors should be included in a proper estimation of environmental impact. Thus in practice, there are net CO_2 emissions resulting from these experiments, but accurate estimation is nontrivial. Future good practice should use emission estimation tools such as CodeCarbon (Schmidt et al., 2021), which can calculate estimated energy usage and emissions by tracking compute. Regrettably, we were not aware of such tools during our experiments, but shall adopt them in future work, and encourage the community to do the same.

The work in this thesis was also supported by air travel between London and Florence, New York and Hong Kong, which is estimated at 4.85 tonnes of effective CO_2 including radiative forcing. This was mitigated by 15 tonnes of carbon offset credits conforming to the ISO 14064 and GHG Emissions Protocol Standards.

Appendix C

Appendices for Unsupervised Reading Comprehension by Cloze Translation

C.1 Cloze Featurization and Translation

Cloze questions are featurized as follows. Assume we have a cloze question extracted from a paragraph “the Paris Sevens became the last stop on the calendar in ____.”, and the answer “2018”. We first tokenize the cloze question, and discard it if it is longer than 40 tokens. We then replace the “blank” with a special mask token. If the answer was extracted using the noun phrase chunker, there is no specific answer entity typing so we just use a single mask token “[MASK]”. However, when we use the named entity answer generator, answers have a named entity label, which we can use to give the cloze translator a high level idea of the answer semantics. In the example above, the answer “2018” has the named entity type “DATE”. We group fine grained entity types into higher level categories, each with its own masking token as shown in Table 3.1, and so the mask token for this example is “[TEMPORAL]”.

C.2 Unsupervised NMT Training Setup Details

We use the English tokenizer from Moses (Koehn et al., 2007), and use FastBPE (https://github.com/glample/fastBPE) to split into subword units, with a vocabulary size of 60000. The architecture uses a 4-layer transformer encoder and 4-layer transformer decoder, where one layer is language specific for both the encoder and decoder, and the rest are shared. We use the standard hyperparameter settings recommended by Lample et al. (2018c). The models are initialised with random weights, and the input word embedding matrix is initialised using FastText vectors (Bojanowski et al., 2017) trained on the concatenation of the $\mathcal{D}_s$ and $\mathcal{D}_t$ corpora. Initially, the auto-encoding loss and back-translation loss have equal weight, with the auto-encoding loss coefficient reduced to 0.1 by 100K steps and to 0 by 300k steps. We train using 5M cloze questions and natural questions, and cease training when the BLEU scores between back-translated and input questions stops improving, usually around 300K optimisation steps. When generating, we decode greedily, and note that decoding with a beam size of 5 did not significantly change downstream QA performance, or greatly change the fluency of generations.

C.3 RC Model Training Details

We train BiDAF + Self Attention using the default settings. We evaluate using a synthetic development set of data generated from 1000 context paragraphs every 500 training steps, and halt when the performance has not changed by 0.1% for the last 5 evaluations. We train BERT-Base and BERT-Large with a batch size of 16, and the default learning rate hyperparameters. For BERT-Base, we evaluate using a synthetic development set of data generated from 1000 context paragraphs every 500 training steps, and halt when the performance has not changed by 0.1% for the last 5 evaluations. For BERT-Large, due to larger model size, training takes longer, so we manually halt training when the synthetic development set performance plateaus, rather than using the automatic early stopping.

C.4 Language Model Pretraining

We experimented with Masked Language Model (MLM) pretraining of the translation models, $p_{s \to t}(q|q')$ and $p_{t \to s}(q'|q)$. We use the XLM implementation (https://github.com/facebookresearch/XLM) and use default hyperparameters for both MLM pretraining and UMT fine-tuning. The UMT encoder is initialised with the MLM model's parameters, and the decoder is randomly initialised. We find translated questions to be qualitatively more fluent and abstractive than the those from the models used in the main experiments. The well-formedness model classified 78.5 of this model's generated generations as well-formed, 10.5% higher than without pretraining, surpassing the rule-based question generator of Heilman and Smith (2010). However, using MLM pretraining did not lead to significant differences for question answering performance (the main focus of this chapter), so we leave a thorough investigation into language model pretraining for unsupervised question answering as future work.

C.5 More Examples of Unsupervised MT Cloze Translations

Table C.1 shows many more examples of cloze question translations from our best-performing UMT approach (sub-clause boundaries and wh* heuristics).

Cloze Question	Answer	Generated Question
to record their sixth album in TEMPORAL	2005	When will they record their sixth album ?
Redline management got word that both were negotiating with THING	Trek/Gary Fisher	What Redline management word got that both were negotiating ?
Reesler to suspect that Hitchin murdered PERSON/NORP/ORG	Wright	Who is Reesler to suspect that Hitchin murdered ?
joined PERSON/NORP/ORG in the 1990s to protest the Liberals' long-gun registry	the Reform Party	Who joined in the 1990s to protest the Liberals ' long-gun registry ?
to end the TEMPORAL NLCS, and the season, for the New York Mets	2006	When will the NLCS end , and the season , for the New York Mets ?
NUMERIC of the population concentrated in the province of Lugo	about 75%	How many of you are concentrated in the province of Lugo ?
placed NUMERIC on uneven bars and sixth on balance beam	fourth	How many bars are placed on uneven bars and sixth on balance beam ?
to open a small branch in PLACE located in Colonia Escalon in San Salvador	La Casona	Where do I open a small branch in Colonia Escalon in San Salvador ?
they finished outside the top eight when considering only THING events	World Cup	What if they finished outside the top eight when considering only events ?
he obtained his Doctor of Law degree in 1929.Who's who in PLACE	America	Where can we obtain our Doctor of Law degree in 1929.Who ' s who ?
to establish the renowned Paradise Studios in PLACE in 1979	Sydney	Where is the renowned Paradise Studios in 1979 ?
Ukraine came out ahead NUMERIC	four to three	How much did Ukraine come out ahead ?
their rule over these disputed lands was cemented after another Polish victory, in THING	the Polish-Soviet War	What was their rule over these disputed lands after another Polish victory , anyway ?
sinking PERSON/NORP/ORG 35 before being driven down by depth charge attacks	Patrol Boat	Who is sinking 35 before being driven down by depth charge attacks ?
to hold that PLACE was the sole or primary perpetrator of human rights abuses	North Korea	Where do you hold that was the sole or primary perpetrator of human rights abuses ?
to make it 2–1 to the Hungarians, though PLACE were quick to equalise	Italy	Where do you make it 2-1 to the Hungarians , though quick equalise ?
he was sold to Colin Murphy's Lincoln City for a fee of £NUMERIC	15,000	How much do we need Colin Murphy ' s Lincoln City for a fee ?
Bierut is the co-founder of the blog PERSON/NORP/ORG	Design Observer	Who is the Bierut co-founder of the blog ?
the Scotland matches at the 1982 THING being played in a "family atmosphere"	FIFA World Cup	What are the Scotland matches at the 1982 being played in a " family atmosphere " ?
Tom realizes that he has finally conquered both "THING" and his own stage fright	La Cinquette	What happens when Tom realizes that he has finally conquered both " and his own stage fright ?
it finished first in the PERSON/NORP/ORG ratings in April 1990	Arbitron	Who finished it first in the ratings in April 1990 ?
his observer to destroy NUMERIC others	two	How many others can his observer destroy ?
Martin had recorded some solo songs (including "Never Back Again") in 1984 in PLACE	the United Kingdom	Where have Martin recorded some solo songs (including " Never Back Again ") in 1984 ?
the NUMERIC occurs under stadium lights	second	How many lights occurs under stadium ?
PERSON/NORP/ORG had made a century in the fourth match	Poulton	Who had made a century in the fourth match ?
mentions the Bab and THING	Bábís	What are the mentions of Bab ?

Table C.1: Further cloze translations from the UMT model (with sub-clause boundaries and wh* heuristic applied)

Appendix D

Appendices for Evaluating Cross-Lingual Reading Comprehension

D.1 Further details on Parallel Sentence mining

Table D.1 shows the number of mined parallel sentences found in each language, as function of how many languages the sentences are parallel between. As the number of languages that a parallel sentence is shared between increases, the number of such sentences decreases. When we look for 7-way aligned examples, we only find 1340 sentences from the entirety of the 7 Wikipedias. Additionally, most of these sentences are the first sentence of the article, or are uninteresting. However, if we choose 4-way parallel sentences, there are plenty of sentences to choose from. We sample evenly from each combination of English and 3 of the 6 target languages. This ensures that we have an even distribution over all the target languages, as well as ensuring we have even numbers of instances that will be parallel between target language combinations.

N-way	en	de	es	ar	zh	vi	hi
2	12219436	3925542	4957438	1047977	1174359	904037	210083
3	2143675	1157009	1532811	427609	603938	482488	83495
4	385396	249022	319902	148348	223513	181353	34050
5	73918	56756	67383	44684	58814	54884	13151
6	12333	11171	11935	11081	11485	11507	4486
7	1340	1340	1340	1340	1340	1340	1340

Table D.1: Number of mined parallel sentences as a function of how many languages the sentences are parallel between

Appendix E

Appendices for How Context Affects Language Models' Factual Predictions

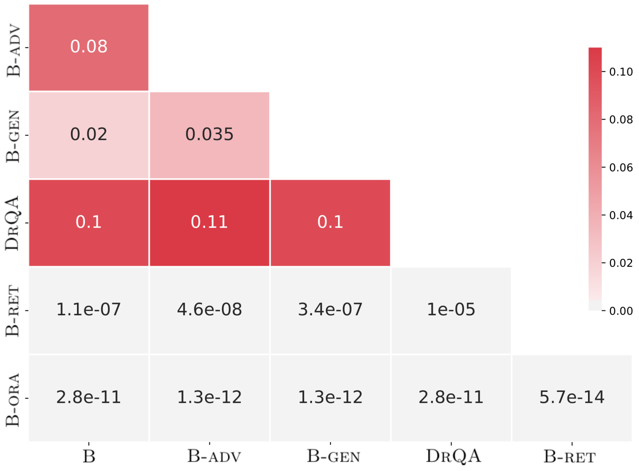


Figure E.1: Pairwise statistical significance for the results presented in Table 5.2.

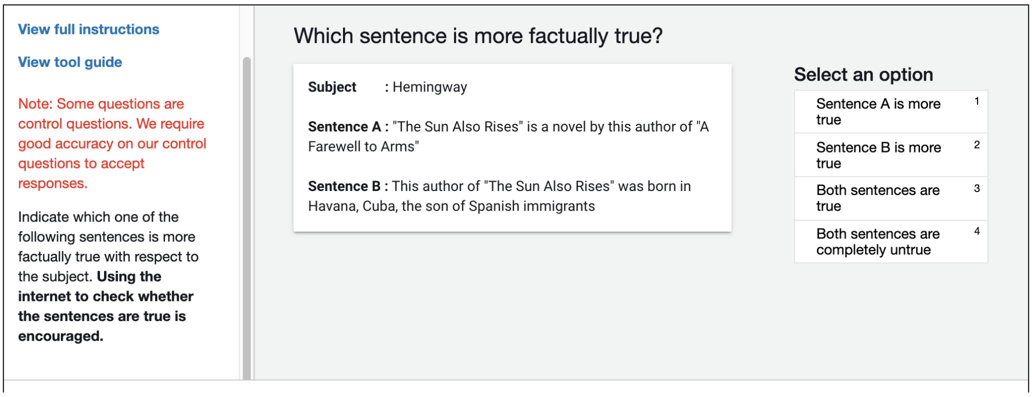
E.1 Statistical Significance Tests

Figure E.1 shows pairwise statistical significance for the results presented in Table 5.2, using the sign test across relations. Each cell reports the p-value of the corresponding pair. The improvements achieved by B-RET and B-ORA are statistically significant (p-value $<$ than the alpha level of 0.05)

Appendix F

Appendices for Retrieval-Augmented Generation for Knowledge-Intensive NLP

F.1 Human Evaluation


View full instructions
View tool guide
Select an option
Sentence A is more true 1
Sentence B is more true 2
Both sentences are true 3
Both sentences are completely untrue 4

Note: Some questions are control questions. We require good accuracy on our control questions to accept responses.

Indicate which one of the following sentences is more factually true with respect to the subject. **Using the internet to check whether the sentences are true is encouraged.**

Which sentence is more factually true?

Subject : Hemingway

Sentence A : "The Sun Also Rises" is a novel by this author of "A Farewell to Arms"

Sentence B : This author of "The Sun Also Rises" was born in Havana, Cuba, the son of Spanish immigrants

Figure F.1: Annotation interface for human evaluation of factuality. A pop-out for detailed instructions and a worked example appear when clicking "view tool guide".

Figure F.1 shows the interface for human evaluation. To avoid biases from screen position, the model corresponding to sentence A and B was randomly chosen for each example. Annotators were encouraged to research the topic using the internet, and given detailed instructions and worked examples. We included a number of control gold examples to assess the accuracy of the annotators. Two annotators did

Task	Train	Development	Test
Jeopardy Question Generation	97392	13714	26849
FEVER-3-way	145450	10000	10000
FEVER-2-way	96966	6666	6666

Table F.1: Number of instances in the datasets, where not already covered in Table 2.2.

not perform well on the control examples and their annotations were removed.

F.2 Further Details on Open-Domain QA

For TriviaQA, there are often many valid answer references for a question, some of which are not suitable training targets, such as emoji, so we filter out answer candidates if they do not occur in top 1000 documents for the query. The answers for CT are given as regular expressions, which are not suitable generation targets. We use a pre-processing step where we first retrieve the top 1000 documents for each query, and take the most frequently-matched answer as the supervision target. If no matches are found, we use a simple heuristic: generate all possible permutations of the regex, replacing non-deterministic symbols with whitespace.

F.3 Parameters

Our RAG models contain trainable parameters for the BERT-base query and document encoders of DPR, 110M parameters each, and 406M trainable parameters from BART-large, 406M parameters, making a total of 626M trainable parameters. The non-parametric memory index does not consist of trainable parameters, but does consists of 21M 728-dim vectors, consisting of 15.3B floats. Subsequent work has demonstrated that dimensions can be reduced to around 200 dimensions without loss of accuracy, which would represent an index with 4.2B floats.

F.4 Number of Instances per Dataset

The number of training, development and test instances for Jeopardy and FEVER are shown in Table F.1. Statistics for other datasets can be found in Table 2.2.

Appendix G

Appendices for Question And Answer Test-Train Overlap in Open-Domain QA Datasets

G.1 Additional Question Overlap Examples

Tables G.1, G.3 and G.2 give more question overlap examples for the three datasets.

Answer	Test Question	Train Question
Bob Geldof	who played pink in pink floyd the wall	who played pink in the movie the wall
Daren Maxwell Kagasoff	who played ricky in secret life of the american teenager	who played ricky on the secret life of the american teenager
Andy	who does april end up with on parks and rec	who does april marry in parks and rec
may 5 2017	when did gaurdians of the galaxy 2 come out	when is guardians of the galaxy vol 2 released
norman pritchard	who won the first medal in olympics for india	who won the first individual olympic medal for india
moira kelly	who does the voice of nala in the lion king	who played nala in the lion king movie
supreme court	who enforces the charter of rights and freedoms	who has final authority over the canadian charter of rights & freedoms
554	most passing yards by nfl qb in a game	what is the nfl record for most passing yards in a single game
John Ross	who ran the fastest 40 yard dash in the nfl	who has the fastest 40 yard dash ever
international border ib	what is the name of india pakistan border	what is the border name between india and pakistan
Andrew Wright	who wrote when a man loves a woman	who wrote song when a man loves a woman
new england patriots	who has participated in the most super bowls	what nfl team has been to most super bowls

Table G.1: Additional examples of test-train overlapping questions in Natural Questions

Answer	Test Question	Train Question
costa rica	where is isthmus of panama located on the map?	where is isthmus of panama located?
1986 world series	when’s the last time the mets won the world series?	when did the mets win the pennant?
abbottabad	where was bin laden found and killed?	what country was osama bin laden killed in?
believer	what other movies has ryan gosling been in?	what movies does ryan gosling star in?
sculpture	what type of art did leonardo da vinci make?	what kind of art did leonardo da vinci produce?
origin of species	what book did charles darwin wrote in 1859?	what was the name of the book that charles darwin wrote?
morehouse college	what college did martin luther king jr go to?	where did dr. martin luther king jr. go to school?
communist state	what type of government did soviet union have?	what type of government does the former soviet union have?
turkish lira	what money to take to turkey?	what currency to take to side turkey?
spanish language	what is the most common language spoken in argentina?	what is language in argentina?
opera OR classical music	what music period did beethoven live in?	what music did beethoven composed?
harry s truman	who was president after franklin d. roosevelt?	who became president when roosevelt died in office?

Table G.2: Examples of test-train overlapping questions in WebQuestions

Answer	Test Question	Train Question
Picasso	Who painted ”Boy With a Pipe” which, in May 2004, was sold for a record price of $104 million?	painted in 1905, the painting garcon a la pipe was a famous painting by which famous artist who died in 1973?
Wensum	On what river is the city of Norwich	the english city of norwich lies on which river?
Mantle	Comprising around two-thirds of the Earth’s mass , what is found between the core of the Earth and its crust?	what do we call the layer of the earth between its crust and its core?
Live and Let Die	In which James Bond film does actress Jane Seymour play Solitaire?	jane seymour played the character ”solitaire” in which bond film?
Esau	Who, in the Bible, was the eldest son of Isaac?	in the bible, who was the first born of isaac?
Alanis Morrisette	Who made the 1995 album ’Jagged Little Pill’ which sold 33 million copies?	who released the 1995 hit album ”jagged little pill”?
Excalibur	In British legend, what is the name of King Arthur’s sword?	what was the name of king arthur’s sword?
Humidity	What is measured by a Hygrometer?	what does a hygrometer measure?
A Storm	On the Beaufort scale what is defined as force 11?	what is force 11 (eleven) on the beaufort scale?
Jeremy Irons	Actress Sinead Cusack is married to which ’Oscar’ winning actor?	which actor is the husband of sinead cusack?
Sir Cloudesley Shovell	Who was the British Admiral who died in 1707 when four of his ships were wrecked in the Scilly Isles?	in 1707 a fleet of navy ships was wrecked off the scilly islands. who was the commander who lost his life in the disaster?
Tony Hart	Which famous individual created the ’Blue Peter’ sailing ship logo?	which artist designed the logo for uk television children’s show ‘blue peter’?

Table G.3: Examples of test-train overlapping questions in TriviaQA

Appendix H

Appendices for 65 Million Probably-asked Questions and What You Can Do With Them

H.1 Further Details on Passage Selection

The passage selection model is based on RoBERTa$_{\text{BASE}}$ (Liu et al., 2019c). We feed each passage into the model and use an MLP on top of the [CLS] representation to produce a score. We use this model to obtain a score for every passage in the corpus. The top N highest-scoring passages are selected for QA-pair generation. This model achieves 84.7% recall on the NQ dev set.

H.2 Further Details on Question Quality

For NQ, we find that the retrieved questions are paraphrases of the test questions in the majority of cases. We conduct human evaluation on 50 random sampled questions generated from the Wikipedia passage pool. We make the following observations: i) 82% of questions accurately capture the context of the answer in the passage, and contain sufficient details to locate the answer. ii) 16% of questions have incorrect semantics with respect to their answers. These errors are driven by two main factors: *Mistaking extremely similar entities* and *Generalisation to rare* *phrases*. An example of the former is "what is the eastern end of the Kerch peninsula" for the passage "The Kerch Peninsula is located at the eastern end of the Crimean Peninsula" and the answer "the Crimean Peninsula". An example of the latter is where the model interprets digits separated by colons as date ranges, such as for the passage "under a 109–124 loss to the Milwaukee Bucks", the question is generated as "when did ... play for the Toronto Raptors" iii) only 2% of questions mismatch question wh-words in the analysis sample.

H.3 Further Details on Inference Speed

The machine used for speed benchmarking is a machine learning workstation with 80 CPU cores, 512GB of CPU RAM and access to one 32GB NVIDIA V100 GPU. Inference is carried out at mixed precision for all systems, and questions are allowed to be answered in parallel. Models are implemented in Pytorch (Paszke et al., 2019) using Transformers (Wolf et al., 2020). Measurements are repeated 3 times and the mean time is reported, rounded to an appropriate significant figure. The HNSW index used in this experiment indexes all 65M PAQ QA-pairs with 768 dimensional vectors, uses an ef_construction of 80, ef_search of 32, and store_n of 256, and performs up to 2048 searches in parallel. This index occupies 220GB, but can be considerably compressed with scalar or product quantization, or training retrievers with smaller dimensions – see Section H.6 for details of such an index.

H.4 Further Details on Selective QA

We also investigate improving FiD's calibration on NQ, using a post-hoc calibration technique similar to Jiang et al. (2020a). We train a Gradient Boosting Machine (GBM, Friedman, 2001) on development data to predict whether FiD has answered correctly or not. The GBM is featurised with FiD's answer loss, answer log probability and the retrieval score of the top 100 retrieved documents from DPR. Figure H.1 shows these results. We first note that FiD-Large's answer loss and answer log probabilities perform similarly, and both struggle to calibrate FiD, as mentioned in the main chapter. The GBM improves calibration, especially at lower coverages, but still lags behind RePAQ by 7% EM at 50% coverage. We also note that we can actu-

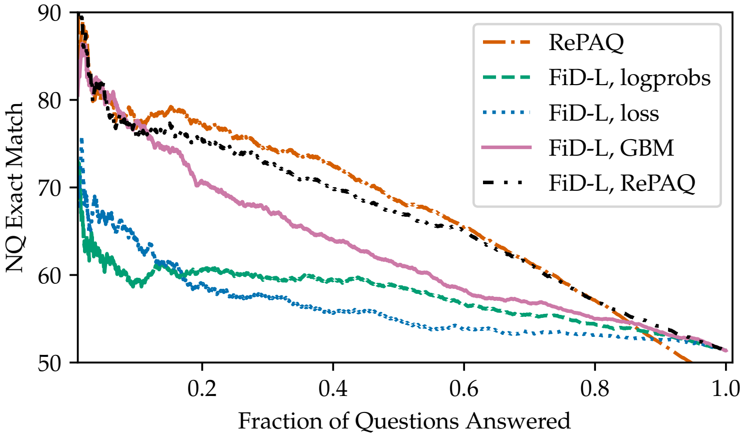


Figure H.1: Risk-Coverage Plot for different calibration methods for FiD (RePAQ included for comparison). Using RePAQ's confidence scores to calibrate FiD leads to FiD's strongest results

ally use RePAQ's confidence scores to calibrate FiD. Here, we use FiD's predicted answer, but RePAQ's confidence score to decide whether to answer or not. This result is also plotted in Figure H.1, and results in FiD's best risk-coverage curve. Despite these improvements, FiD is still not as well-calibrated as RePAQ.

H.5 Additional Model training details

RePAQ models were trained for up to 3 days on a machine with 8 NVIDIA 32GB V100 GPUs. Validation Exact Match score was used to determine when to stop training in all cases. RePAQ retrievers were trained using Fairseq (Ott et al., 2019), and rerankers were trained in Transformers (Wolf et al., 2020) in Pytorch (Paszke et al., 2019). The PAQ CBQA models were trained in Fairseq for up to 6 days on 8 NVIDIA 32GB V100 GPUs, after which validation accuracy had plateaued. Hyperparameters were tuned to try to promote faster learning, but learning became unstable with learning rates greater than 0.0001.

H.6 Memory-Efficient RePAQ Retriever

As part of the open-source release, we have trained a memory-efficient RePAQ retriever designed for use with more modest hardware than the main RePAQ models. This consists of an ALBERT-base retriever, with 256-dimensional embedding, rather than the 768-dimensional models in the main paper. We provide 2 FAISS indices (Johnson et al., 2019) for use with this model, both built with 8-bit scalar quantization. The first index is a flat index, which is very memory-friendly, requiring only 16GB of CPU RAM, but is relatively slow (1-10 questions per second). The other is an HNSW approximate index (Malkov and Yashunin, 2020), requiring $\sim$32 GB of CPU RAM, but can process 100-1000 questions per second. This memory-efficient system is highly competitive with the models in the main paper, actually outperforming the ALBERT-base model (+0.6%, NQ, +0.5%, TQA), and only trailing the ALBERT-xlarge model by 0.6% on average (-0.3% NQ, -0.9% TQA).

Bibliography

Julia Adams, Hannah Brückner, and Cambria Naslund. 2019. Who Counts as a Notable Sociologist on Wikipedia? Gender, Race, and the "Professor Test". *Socius*, 5:2378023118823946. Publisher: SAGE Publications.

Alan Akbik, Laura Chiticariu, Marina Danilevsky, Yunyao Li, Shivakumar Vaithyanathan, and Huaiyu Zhu. 2015. Generating High Quality Proposition Banks for Multilingual Semantic Role Labeling. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 397–407, Beijing, China. Association for Computational Linguistics.

Rami Al-Rfou, Guillaume Alain, Amjad Almahairi, Christof Angermüller, Dzmitry Bahdanau, Nicolas Ballas, Frédéric Bastien, Justin Bayer, Anatoly Belikov, Alexander Belopolsky, Yoshua Bengio, Arnaud Bergeron, James Bergstra, Valentin Bisson, Josh Bleecher Snyder, Nicolas Bouchard, Nicolas Boulanger-Lewandowski, Xavier Bouthillier, Alexandre de Brébisson, Olivier Breuleux, Pierre Luc Carrier, Kyunghyun Cho, Jan Chorowski, Paul F. Christiano, Tim Cooijmans, Marc-Alexandre Côté, Myriam Côté, Aaron C. Courville, Yann N. Dauphin, Olivier Delalleau, Julien Demouth, Guillaume Desjardins, Sander Dieleman, Laurent Dinh, Melanie Ducoffe, Vincent Dumoulin, Samira Ebrahimi Kahou, Dumitru Erhan, Ziye Fan, Orhan Firat, Mathieu Germain, Xavier Glorot, Ian J. Goodfellow, Matthew Graham, Çaglar Gülçehre, Philippe Hamel, Iban Harlouchet, Jean-Philippe Heng, Balázs Hidasi, Sina Honari, Arjun Jain, Sébastien Jean, Kai Jia, Mikhail Korobov, Vivek Kulkarni, Alex Lamb, Pascal Lamblin, Eric Larsen, César Laurent, Sean Lee, Simon Lefrançois, Simon Lemieux, Nicholas Léonard, Zhouhan Lin, Jesse A. Livezey, Cory Lorenz, Jeremiah Lowin, Qianli Ma, Pierre-Antoine Manzagol, Olivier Mastropietro, Robert McGibbon, Roland Memisevic, Bart van Merriënboer, Vincent Michalski, Mehdi Mirza, Alberto Orlandi, Christopher Joseph Pal, Razvan Pascanu, Mohammad Pezeshki, Colin Raffel, Daniel Renshaw, Matthew Rocklin, Adriana Romero, Markus Roth, Peter Sadowski, John Salvatier, François Savard, Jan Schlüter, John Schulman, Gabriel Schwartz, Iulian Vlad Serban, Dmitriy Serdyuk, Samira Shabanian, Étienne Simon, Sigurd Spieckermann, S. Ramana Subramanyam, Jakub Sygnowski, Jérémie Tanguay, Gijs van Tulder, Joseph P. Turian, Sebastian Urban, Pascal Vincent, Francesco Visin, Harm de Vries, David Warde-Farley, Dustin J. Webb, Matthew Willson, Kelvin Xu, Lijun Xue, Li Yao, Saizheng Zhang, and Ying Zhang. 2016. Theano: A Python framework for fast computation of mathematical expressions. *CoRR*, abs/1605.02688. ArXiv: 1605.02688.

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic QA Corpora Generation with Roundtrip Consistency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6168–6173, Florence, Italy. Association for Computational Linguistics.

Ahmed Aly, Kushal Lakhotia, Shicong Zhao, Mrinal Mohit, Barlas Oguz, Abhinav Arora, Sonal Gupta, Christopher Dewan, Stef Nelson-Lindall, and Rushin Shah. 2018. PyText: A Seamless Path from NLP research to production. *arXiv preprint arXiv:1812.08729*.

Raviteja Anantha, Svitlana Vakulenko, Zhucheng Tu, Shayne Longpre, Stephen Pulman, and Srinivas Chappidi. 2021. Open-Domain Question Answering Goes Conversational via Question Rewriting. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguis-*

tics: Human Language Technologies, pages 520–534, Online. Association for Computational Linguistics.

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. Leveraging Linguistic Structure For Open Domain Information Extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354, Beijing, China. Association for Computational Linguistics.

S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Zitnick, and D. Parikh. 2015. VQA: Visual Question Answering. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2425–2433, Los Alamitos, CA, USA. IEEE Computer Society. ISSN: 2380-7504.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. Unsupervised Statistical Machine Translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3632–3642, Brussels, Belgium. Association for Computational Linguistics.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the Cross-lingual Transferability of Monolingual Representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

Mikel Artetxe and Holger Schwenk. 2019a. Margin-based Parallel Corpus Mining with Multilingual Sentence Embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3197–3203, Florence, Italy. Association for Computational Linguistics.

Mikel Artetxe and Holger Schwenk. 2019b. Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Akari Asai and Eunsol Choi. 2021. Challenges in Information-Seeking QA: Unanswerable Questions and Paragraph Retrieval. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1492–1504, Online. Association for Computational Linguistics.

Akari Asai, Akiko Eriguchi, Kazuma Hashimoto, and Yoshimasa Tsuruoka. 2018. Multilingual Extractive Reading Comprehension by Runtime Machine Translation. *arXiv:1809.03275 [cs]*. ArXiv: 1809.03275.

Akari Asai, Jungo Kasai, Jonathan Clark, Kenton Lee, Eunsol Choi, and Hannaneh Hajishirzi. 2021. XOR QA: Cross-lingual Open-Retrieval Question Answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 547–564, Online. Association for Computational Linguistics.

Giuseppe Attardi. 2015. WikiExtractor. Publication Title: GitHub repository.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. *arXiv:1611.09268 [cs]*. ArXiv: 1611.09268.

Junwei Bao, Nan Duan, Zhao Yan, Ming Zhou, and Tiejun Zhao. 2016. Constraint-Based Question Answering with Knowledge Graph. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2503–2514, Osaka, Japan. The COLING 2016 Organizing Committee.

Max Bartolo, Alastair Roberts, Johannes Welbl, Sebastian Riedel, and Pontus Stenetorp. 2020. Beat the AI: Investigating Adversarial Human Annotation for Reading Comprehension. *Transactions of the Association for Computational Linguistics*, 8:662–678. _eprint: https://doi.org/10.1162/tacl_a_00338.

Petr Baudis and Jan Sedivy. 2015. Modeling of the question answering task in the yodaqa system. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 222–228. Springer.

Yonatan Belinkov and James Glass. 2019. Analysis Methods in Neural Language Processing: A Survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic Parsing on Freebase from Question-Answer Pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA. Association for Computational Linguistics.

Bin Bi, Chenliang Li, Chen Wu, Ming Yan, Wei Wang, Songfang Huang, Fei Huang, and Luo Si. 2020. PALM: Pre-training an Autoencoding&Autoregressive Language Model for Context-conditioned Generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8681–8691, Online. Association for Computational Linguistics.

Ning Bian, Xianpei Han, Bo Chen, Hongyu Lin, Ben He, and Le Sun. 2021. Bridging the Gap between Language Model and Reading Comprehension: Unsupervised MRC via Self-Supervision. *arXiv:2107.08582 [cs]*. ArXiv: 2107.08582.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. PIQA: Reasoning about Physical Commonsense in Natural Language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *J. Mach. Learn. Res.*, 3(null):993–1022. Publisher: JMLR.org.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146. _eprint: https://direct.mit.edu/tacl/article-pdf/doi/10.1162/tacl_a_00051/1567442/tacl_a_00051.pdf.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, SIGMOD '08, pages 1247–1250, Vancouver, Canada. Association for Computing Machinery.

Samuel R. Bowman. 2021. When Combating Hype, Proceed with Caution. *arXiv:2110.08300 [cs]*. ArXiv: 2110.08300.

Eric Brill, Eugene Charniak, Mary Harper, Marc Light, Ellen Riloff, and Ellen Voorhees. 2000. ANLP-NAACL 2000 Workshop: Reading Comprehension Tests as Evaluation for Computer-Based Language Understanding Systems. In *ANLP-NAACL 2000 Workshop: Reading Comprehension Tests as Evaluation for Computer-Based Language Understanding Systems*. Association for Computational Linguistics.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1993. Signature Verification Using a "Siamese" Time Delay Neural Network. In *Proceedings of the 6th International Conference on Neural Information Processing Systems*, NIPS'93, pages 737–744, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. Event-place: Denver, Colorado.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

B. Barla Cambazoglu, Mark Sanderson, Falk Scholer, and Bruce Croft. 2021. A Review of Public Datasets in Question Answering Research. *SIGIR Forum*, 54(2). Place: New York, NY, USA Publisher: Association for Computing Machinery.

William Chan, Nikita Kitaev, Kelvin Guu, Mitchell Stern, and Jakob Uszkoreit. 2019. KERMIT: Generative Insertion-Based Modeling for Sequences. *arXiv:1906.01604 [cs, stat]*. ArXiv: 1906.01604.

Eugene Charniak, Yasemin Altun, Rodrigo de Salvo Braz, Benjamin Garrett, Margaret Kosmala, Tomer Moscovich, Lixin Pang, Changhee Pyo, Ye Sun, Wei Wy, Zhongfa Yang, Shawn Zeiler, and Lisa Zorn. 2000. Reading Comprehension Programs in a Statistical-Language-Processing Class. In *ANLP-NAACL 2000 Workshop: Reading Comprehension Tests as Evaluation for Computer-Based Language Understanding Systems*.

Bo Chen, Bo An, Le Sun, and Xianpei Han. 2018. Semi-Supervised Lexicon Learning for Wide-Coverage Semantic Parsing. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 892–904, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Danqi Chen. 2018. *Neural Reading Comprehension and Beyond*. PhD Thesis, Stanford University.

Danqi Chen, Jason Bolton, and Christopher D. Manning. 2016. A Thorough Examination of the CNN/Daily Mail Reading Comprehension Task. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2358–2367, Berlin, Germany. Association for Computational Linguistics.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to Answer Open-Domain Questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

Danqi Chen and Wen-tau Yih. 2020. Open-Domain Question Answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 34–37, Online. Association for Computational Linguistics.

Hongge Chen, Si Si, Yang Li, Ciprian Chelba, Sanjiv Kumar, Duane Boning, and Cho-Jui Hsieh. 2020. Multi-Stage Influence Function. In *Advances in Neural Information Processing Systems*, volume 33, pages 12732–12742. Curran Associates, Inc.

Xilun Chen, Kushal Lakhotia, Barlas Oğuz, Anchit Gupta, Patrick Lewis, Stan Peshterliev, Yashar Mehdad, Sonal Gupta, and Wen-tau Yih. 2021. Salient Phrase Aware Dense Retrieval: Can a Dense Retriever Imitate a Sparse One? *arXiv:2110.06918 [cs]*. ArXiv: 2110.06918.

Hao Cheng, Yelong Shen, Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2021. UnitedQA: A Hybrid Approach for Open Domain Question Answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3080–3090, Online. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language*

Processing (EMNLP), pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question Answering in Context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184, Brussels, Belgium. Association for Computational Linguistics.

Eunsol Choi, Daniel Hewlett, Jakob Uszkoreit, Illia Polosukhin, Alexandre Lacoste, and Jonathan Berant. 2017. Coarse-to-Fine Question Answering for Long Documents. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 209–220, Vancouver, Canada. Association for Computational Linguistics.

Jennifer Chu-carroll, John Prager, Christopher Welty, Krzysztof Czuba, and David Ferrucci. 2003. A Multi-Strategy and Multi-Source Approach to Question Answering. In *In Proceedings of Text REtrieval Conference*.

Yu-An Chung, Hung-Yi Lee, and James Glass. 2018. Supervised and Unsupervised Transfer Learning for Question Answering. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1585–1594, New Orleans, Louisiana. Association for Computational Linguistics.

Philipp Cimiano, Vanessa López, Christina Unger, Elena Cabrio, Axel-Cyrille Ngonga Ngomo, and Sebastian Walter. 2013. Multilingual Question Answering over Linked Data (QALD-3): Lab Overview. In *CLEF*.

Christopher Clark and Matt Gardner. 2018. Simple and Effective Multi-Paragraph Reading Comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 845–855, Melbourne, Australia. Association for Computational Linguistics.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. TyDi QA: A Benchmark for Information-Seeking Question Answering in Typologically Diverse Languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

Cyril Cleverdon. 1967. The Cranfield Tests on Index Language Devices. *Aslib Proceedings*, 19(6):173–194. Publisher: MCB UP Ltd.

Jacob Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46. Publisher: SAGE Publications Inc.

Ronan Collobert and Jason Weston. 2008. A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 160–167, New York, NY, USA. Association for Computing Machinery. Event-place: Helsinki, Finland.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural Language Processing (Almost) from Scratch. *J. Mach. Learn. Res.*, 12(null):2493–2537. Publisher: JMLR.org.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for*

Computational Linguistics, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual Language Model Pretraining. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating Cross-lingual Sentence Representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2019a. Cross-Lingual Machine Reading Comprehension. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1586–1595, Hong Kong, China. Association for Computational Linguistics.

Yiming Cui, Ting Liu, Wanxiang Che, Li Xiao, Zhipeng Chen, Wentao Ma, Shijin Wang, and Guoping Hu. 2019b. A Span-Extraction Dataset for Chinese Machine Reading Comprehension. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing*. Association for Computational Linguistics.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, and Furu Wei. 2021. Knowledge Neurons in Pretrained Transformers. *arXiv:2104.08696 [cs]*. ArXiv: 2104.08696.

Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. 2018. Embodied Question Answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, and Andrew McCallum. 2019.

Multi-step Retriever-Reader Interaction for Scalable Open-domain Question Answering. In *ICLR (Poster)*. OpenReview.net.

Rajarshi Das, Manzil Zaheer, Dung Thai, Ameya Godbole, Ethan Perez, Jay Yoon Lee, Lizhen Tan, Lazaros Polymenakos, and Andrew McCallum. 2021. Case-based Reasoning for Natural Language Queries over Knowledge Bases. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9594–9611, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2019. Question Answering by Reasoning Across Documents with Graph Convolutional Networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2306–2317, Minneapolis, Minnesota. Association for Computational Linguistics.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021a. Editing Factual Knowledge in Language Models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021b. Autoregressive Entity Retrieval. In *International Conference on Learning Representations*.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407. Publisher: John Wiley & Sons, Ltd.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In

Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Bhuwan Dhingra, Danish Danish, and Dheeraj Rajagopal. 2018. Simple and Effective Semi-Supervised Question Answering. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 582–587, New Orleans, Louisiana. Association for Computational Linguistics.

Martin d'Hoffschmidt, Wacim Belblidia, Quentin Heinrich, Tom Brendlé, and Maxime Vidal. 2020. FQuAD: French Question Answering Dataset. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1193–1208, Online. Association for Computational Linguistics.

Virginia Dignum. 2019. *Responsible Artificial Intelligence: How to Develop and Use AI in a Responsible Way*, 1st edition. Springer Publishing Company, Incorporated.

Emily Dinan, Angela Fan, Adina Williams, Jack Urbanek, Douwe Kiela, and Jason Weston. 2020. Queens are Powerful too: Mitigating Gender Bias in Dialogue Generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8173–8188, Online. Association for Computational Linguistics.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of Wikipedia: Knowledge-Powered Conversational Agents. In *International Conference on Learning Representations*.

Xinya Du and Claire Cardie. 2018. Harvesting Paragraph-level Question-Answer Pairs from Wikipedia. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1907–1917, Melbourne, Australia. Association for Computational Linguistics.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to Ask: Neural Question Generation for Reading Comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1342–1352, Vancouver, Canada. Association for Computational Linguistics.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A Reading Comprehension Benchmark Requiring Discrete Reasoning Over Paragraphs. In *NAACL-HLT*.

Matthew Dunn, Levent Sagun, Mike Higgins, V. Ugur Guney, Volkan Cirik, and Kyunghyun Cho. 2017. SearchQA: A New Q&A Dataset Augmented with Context from a Search Engine. *arXiv:1704.05179 [cs]*. ArXiv: 1704.05179.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-REx: A Large Scale Alignment of Natural Language with Knowledge Base Triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Alexander Fabbri, Patrick Ng, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2020. Template-Based Question Generation from Retrieved Sentences for Improved Unsupervised Question Answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4508–4513, Online. Association for Computational Linguistics.

Martin Fajcik, Martin Docekal, Karel Ondrej, and Pavel Smrz. 2021. R2-D2: A Modular Baseline for Open-Domain Question Answering. *arXiv:2109.03502 [cs]*. ArXiv: 2109.03502.

Angela Fan, Claire Gardent, Chloé Braud, and Antoine Bordes. 2021. Augmenting Transformers with KNN-Based Composite Memory for Dialog. *Transactions of the Association for Computational Linguistics*, 9:82–99.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long Form Question Answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical Neural Story Generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Yuwei Fang, Shuohang Wang, Zhe Gan, Siqi Sun, and Jingjing Liu. 2020. Accelerating Real-Time Question Answering via Question Generation. *arXiv:2009.05167 [cs]*. ArXiv: 2009.05167.

Manaal Faruqui and Dipanjan Das. 2018. Identifying Well-formed Natural Language Questions. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 798–803, Brussels, Belgium. Association for Computational Linguistics.

William Fedus, Barret Zoph, and Noam Shazeer. 2021. Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. *CoRR*, abs/2101.03961. ArXiv: 2101.03961.

David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A. Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John Prager, Nico Schlaefer, and Chris Welty. 2010. Building Watson: An Overview of the DeepQA Project. *AI Magazine*, 31(3):59–79.

Anjalie Field, Chan Young Park, and Yulia Tsvetkov. 2021. Controlled Analyses of Social Biases in Wikipedia Bios. *CoRR*, abs/2101.00078. ArXiv: 2101.00078.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. MRQA 2019 Shared Task: Evaluating Generalization in Reading Comprehension. In *Proceedings of the 2nd Workshop on Machine Reading for Question*

Answering, pages 1–13, Hong Kong, China. Association for Computational Linguistics.

Nicholas FitzGerald, Julian Michael, Luheng He, and Luke Zettlemoyer. 2018. Large-Scale QA-SRL Parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2051–2060, Melbourne, Australia. Association for Computational Linguistics.

Jerome H. Friedman. 2001. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5):1189–1232. Publisher: Institute of Mathematical Statistics.

Thibault Févry, Livio Baldini Soares, Nicholas FitzGerald, Eunsol Choi, and Tom Kwiatkowski. 2020. Entities as Experts: Sparse Memory Access with Entity Supervision. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4937–4951, Online. Association for Computational Linguistics.

Amir Gandomi and Murtaza Haider. 2015. Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management*, 35(2):137–144.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2021a. The Pile: An 800GB Dataset of Diverse Text for Language Modeling. *CoRR*, abs/2101.00027. ArXiv: 2101.00027.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021b. Making Pre-trained Language Models Better Few-shot Learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.

Matt Gardner, Jonathan Berant, Hannaneh Hajishirzi, Alon Talmor, and Sewon Min.

2019. Question Answering is a Format; When is it Useful? *arXiv:1909.11291 [cs]*. ArXiv: 1909.11291.

Albert Gatt and Emiel Krahmer. 2018. Survey of the State of the Art in Natural Language Generation: Core Tasks, Applications and Evaluation. *J. Artif. Int. Res.*, 61(1):65–170. Place: El Segundo, CA, USA Publisher: AI Access Foundation.

Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen-tau Yih, and Michel Galley. 2018. A Knowledge-Grounded Neural Conversation Model. In *AAAI Conference on Artificial Intelligence*.

Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, and Alfio Gliozzo. 2021. Robust Retrieval Augmented Generation for Zero-shot Slot Filling. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1939–1949, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yoav Goldberg. 2019. Assessing BERT's Syntactic Abilities. *CoRR*, abs/1901.05287. ArXiv: 1901.05287.

David Golub, Po-Sen Huang, Xiaodong He, and Li Deng. 2017. Two-Stage Synthesis Networks for Transfer Learning in Machine Comprehension. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 835–844.

Maharshi Gor, Kellie Webster, and Jordan Boyd-Graber. 2021. Toward Deconfounding the Effect of Entity Demographics for Question Answering Accuracy. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5457–5473, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Bert F. Green, Alice K. Wolf, Carol Chomsky, and Kenneth Laughery. 1961. Baseball: An Automatic Question-Answerer. In *Papers Presented at the May 9-11, 1961, Western Joint IRE-AIEE-ACM Computer Conference*, IRE-AIEE-ACM '61

(Western), pages 219–224, New York, NY, USA. Association for Computing Machinery. Event-place: Los Angeles, California.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor O.K. Li. 2018. Search engine guided neural machine translation. In *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, 32nd AAAI Conference on Artificial Intelligence, AAAI 2018, pages 5133–5140. AAAI press.

Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. 2016. A Deep Relevance Matching Model for Ad-Hoc Retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, pages 55–64, New York, NY, USA. Association for Computing Machinery. Event-place: Indianapolis, Indiana, USA.

Zhaochen Guo and Denilson Barbosa. 2018. Robust named entity disambiguation with random walks. *Semantic Web*, Preprint(Preprint):1–21.

Deepak Gupta, Surabhi Kumari, Asif Ekbal, and Pushpak Bhattacharyya. 2018. MMQA: A Multi-domain Multi-lingual Question-Answering Framework for English and Hindi. In *LREC*.

Kelvin Guu, Tatsunori B. Hashimoto, Yonatan Oren, and Percy Liang. 2018. Generating Sentences by Editing Prototypes. *Transactions of the Association for Computational Linguistics*, 6:437–450.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Retrieval Augmented Language Model Pre-Training. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR.

Momchil Hardalov, Ivan Koychev, and Preslav Nakov. 2019. Beyond English-Only Reading Comprehension: Experiments in Zero-shot Multilingual Transfer for Bulgarian. In *Proceedings of the International Conference on Recent Advances in*

Natural Language Processing (RANLP 2019), pages 447–459, Varna, Bulgaria. INCOMA Ltd.

Tatsunori B Hashimoto, Kelvin Guu, Yonatan Oren, and Percy S Liang. 2018. A Retrieve-and-Edit Framework for Predicting Structured Outputs. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 10052–10062. Curran Associates, Inc.

Wei He, Kai Liu, Jing Liu, Yajuan Lyu, Shiqi Zhao, Xinyan Xiao, Yuan Liu, Yizhong Wang, Hua Wu, Qiaoqiao She, Xuan Liu, Tian Wu, and Haifeng Wang. 2018. DuReader: a Chinese Machine Reading Comprehension Dataset from Real-world Applications. In *Proceedings of the Workshop on Machine Reading for Question Answering*, pages 37–46, Melbourne, Australia. Association for Computational Linguistics.

Michael Heilman and Noah A. Smith. 2010. Good Question! Statistical Ranking for Question Generation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 609–617, Stroudsburg, PA, USA. Association for Computational Linguistics.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching Machines to Read and Comprehend. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1693–1701. Curran Associates, Inc.

Danny Hernandez and Tom B. Brown. 2020. Measuring the Algorithmic Efficiency of Neural Networks. *CoRR*, abs/2005.04305. ArXiv: 2005.04305.

John Hewitt and Christopher D. Manning. 2019. A Structural Probe for Finding Syntax in Word Representations. In *NAACL-HLT (1)*, pages 4129–4138. Association for Computational Linguistics.

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2016. The Goldilocks Principle: Reading Children's Books with Explicit Memory Representations. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Lynette Hirschman, Marc Light, Eric Breck, and John D. Burger. 1999. Deep Read: A Reading Comprehension System. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 325–332, College Park, Maryland, USA. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust Disambiguation of Named Entities in Text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 782–792, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Thomas Hofmann. 1999. Probabilistic Latent Semantic Indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 50–57, New York, NY, USA. Association for Computing Machinery. Event-place: Berkeley, California, USA.

Giwon Hong, Junmo Kang, Doyeon Lim, and Sung-Hyon Myaeng. 2020. Handling Anomalies of Synthetic Questions in Unsupervised Question Answering. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3441–3448, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Matthew Honnibal, Ines Montani, Matthew Honnibal, Henning Peters, Sofie Van Landeghem, Maxim Samsonov, Jim Geovedi, Jim Regan, György Orosz, Søren Lind Kristiansen, Paul O'Leary McCann, Duygu Altinok, Roman, Grégory

Howard, Sam Bozek, Explosion Bot, Mark Amery, Wannaphong Phatthiyaphaibun, Leif Uwe Vogelsang, Björn Böing, Pradeep Kumar Tippa, jeannefukumaru, GregDubbin, Vadim Mazaev, Ramanan Balakrishnan, Jens Dahl Møllerhøj, wbwseeker, Magnus Burton, thomasO, and Avadh Patel. 2019. explosion/spaCy: v2.1.7: Improved evaluation, better language factories and bug fixes.

Tom Hosking and Sebastian Riedel. 2019. Evaluating Rewards for Question Generation Models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2278–2283, Minneapolis, Minnesota. Association for Computational Linguistics.

Nabil Hossain, Marjan Ghazvininejad, and Luke Zettlemoyer. 2020. Simple and Effective Retrieve-Edit-Rerank Text Generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2532–2538, Online. Association for Computational Linguistics.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: the 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, NAACL-Short '06, pages 57–60, USA. Association for Computational Linguistics.

Tsung-Yuan Hsu, Chi-Liang Liu, and Hung-yi Lee. 2019. Zero-shot Reading Comprehension by Cross-lingual Transfer Learning with Multi-lingual Language Representation Model. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5935–5942, Hong Kong, China. Association for Computational Linguistics.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalisation. In *ICML*, pages 4411–4421.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning Deep Structured Semantic Models for Web Search Using Clickthrough Data. In *Proceedings of the 22nd ACM International Conference on Information &amp; Knowledge Management*, CIKM '13, pages 2333–2338, New York, NY, USA. Association for Computing Machinery. Event-place: San Francisco, California, USA.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2020. Poly-encoders: Architectures and Pre-training Strategies for Fast and Accurate Multi-sentence Scoring. In *International Conference on Learning Representations*.

Srinivasan Iyer, Sewon Min, Yashar Mehdad, and Wen-tau Yih. 2021. RECONSIDER: Improved Re-Ranking using Span-Focused Cross-Attention for Open Domain Question Answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1280–1287, Online. Association for Computational Linguistics.

Gautier Izacard and Edouard Grave. 2021a. Distilling Knowledge from Reader to Retriever for Question Answering. In *International Conference on Learning Representations*.

Gautier Izacard and Edouard Grave. 2021b. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

Gautier Izacard, Fabio Petroni, Lucas Hosseini, Nicola De Cao, Sebastian Riedel, and Edouard Grave. 2020. A Memory Efficient Baseline for Open Domain Question Answering. *arXiv:2012.15156 [cs]*. ArXiv: 2012.15156.

Yangfeng Ji, Antoine Bosselut, Thomas Wolf, and Asli Celikyilmaz. 2020. The

Amazing World of Neural Language Generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, pages 37–42, Online. Association for Computational Linguistics.

Robin Jia and Percy Liang. 2017. Adversarial Examples for Evaluating Reading Comprehension Systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020a. How Can We Know What Language Models Know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Zhengbao Jiang, Wei Xu, Jun Araki, and Graham Neubig. 2020b. Generalizing Natural Language Analysis through Span-relation Representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2120–2133, Online. Association for Computational Linguistics.

Yimin Jing, Deyi Xiong, and Zhen Yan. 2019. BiPaR: A Bilingual Parallel Dataset for Multilingual and Cross-lingual Reading Comprehension on Novels. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2452–2462, Hong Kong, China. Association for Computational Linguistics.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, pages 1–1.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Vidur Joshi, Matthew Peters, and Mark Hopkins. 2018. Extending a Parser to Distant Domains Using a Few Dozen Partially Annotated Examples. *arXiv:1805.06556 [cs]*. ArXiv: 1805.06556.

Armand Joulin and Tomas Mikolov. 2015. Inferring Algorithmic Patterns with Stack-Augmented Recurrent Nets. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pages 190–198, Cambridge, MA, USA. MIT Press. Event-place: Montreal, Canada.

H Jégou, M Douze, and C Schmid. 2011. Product Quantization for Nearest Neighbor Search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):117–128.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Nora Kassner and Hinrich Schütze. 2019. Negated LAMA: Birds cannot fly. *arXiv preprint arXiv:1911.03343*.

Nora Kassner, Oyvind Tafjord, Hinrich Schütze, and Peter Clark. 2021. BeliefBank: Adding Memory to a Pre-Trained Language Model for a Systematic Notion of Belief. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8849–8861, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Divyansh Kaushik and Zachary C. Lipton. 2018. How Much Reading Does Reading Comprehension Require? A Critical Investigation of Popular Benchmarks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5010–5015, Brussels, Belgium. Association for Computational Linguistics.

Omri Keren and Omer Levy. 2021. ParaShoot: A Hebrew Question Answering Dataset. In *Proceedings of the 3rd Workshop on Machine Reading for Question Answering*, pages 106–112, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through Memorization: Nearest Neighbor Language Models. In *International Conference on Learning Representations*.

Eugene Kharitonov, Marco Baroni, and Dieuwke Hupkes. 2021. How BPE Affects Memorization in Transformers. *arXiv:2110.02782 [cs]*. ArXiv: 2110.02782.

Omar Khattab, Christopher Potts, and Matei Zaharia. 2021. Relevance-guided Supervision for OpenQA with ColBERT. *Transactions of the Association for Computational Linguistics*, 9:929–944. _eprint: https://direct.mit.edu/tacl/article-pdf/doi/10.1162/tacl_a_00405/1962458/tacl_a_00405.pdf.

Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 39–48, Virtual Event China. ACM.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. 2021. Dynabench: Rethinking Benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4110–4124, Online. Association for Computational Linguistics.

Najoung Kim, Roma Patel, Adam Poliak, Patrick Xia, Alex Wang, Tom McCoy, Ian Tenney, Alexis Ross, Tal Linzen, Benjamin Van Durme, Samuel R. Bowman, and Ellie Pavlick. 2019. Probing What Different NLP Tasks Teach Machines about Function Word Comprehension. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 235–249, Minneapolis, Minnesota. Association for Computational Linguistics.

Walter Kintsch and Teun A. van Dijk. 1978. Toward a model of text comprehension and production. *Psychological Review*, 85(5):363–394. Place: US Publisher: American Psychological Association.

Yuval Kirstain, Patrick Lewis, Sebastian Riedel, and Omer Levy. 2021. A Few More Examples May Be Worth Billions of Parameters. *arXiv:2110.04374 [cs]*. ArXiv: 2110.04374.

Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. 2012. Inducing Crosslingual Distributed Representations of Words. In *Proceedings of COLING 2012*, pages 1459–1474, Mumbai, India. The COLING 2012 Organizing Committee.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180. Association for Computational Linguistics. Event-place: Prague, Czech Republic.

Pang Wei Koh and Percy Liang. 2017. Understanding Black-box Predictions via Influence Functions. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1885–1894. PMLR. ISSN: 2640-3498.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, Red Hook, NY, USA. Curran Associates Inc. Event-place: Lake Tahoe, Nevada.

Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask Me Anything: Dynamic Memory Networks for Natural Language Processing. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 1378–1387. JMLR.org. Event-place: New York, NY, USA.

Vishwajeet Kumar, Nitish Joshi, Arijit Mukherjee, Ganesh Ramakrishnan, and Preethi Jyothi. 2019. Cross-Lingual Training for Automatic Question Generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4863–4872, Florence, Italy. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: a Benchmark for Question Answering Research. *Transactions of the Association of Computational Linguistics*, 7:452–466.

Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. 2019. Quantifying the Carbon Emissions of Machine Learning. *arXiv:1910.09700 [cs]*. ArXiv: 1910.09700.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018a. Unsupervised Machine Translation Using Monolingual Corpora Only. In *International Conference on Learning Representations*.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018b. Word translation without parallel data. In *International Conference on Learning Representations*.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018c. Phrase-Based & Neural Unsupervised Machine

Translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5039–5049, Brussels, Belgium. Association for Computational Linguistics.

Guillaume Lample, Alexandre Sablayrolles, Marc' Aurelio Ranzato, Ludovic Denoyer, and Herve Jegou. 2019. Large Memory Layers with Product Keys. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8548–8559. Curran Associates, Inc.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *International Conference on Learning Representations*.

Angeliki Lazaridou, Adhiguna Kuncoro, Elena Gribovskaya, Devang Agrawal, Adam Liska, Tayfun Terzi, Mai Gimenez, Cyprien de Masson d'Autume, Sebastian Ruder, Dani Yogatama, Kris Cao, Tomás Kociský, Susannah Young, and Phil Blunsom. 2021. Pitfalls of Static Language Modelling. *CoRR*, abs/2102.01951. ArXiv: 2102.01951.

Teven Le Scao and Alexander Rush. 2021. How many data points is a prompt worth? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2627–2636, Online. Association for Computational Linguistics.

Chia-Hsuan Lee and Hung-Yi Lee. 2019. Cross-Lingual Transfer Learning for Question Answering. *arXiv:1907.06042 [cs]*. ArXiv: 1907.06042.

Jinhyuk Lee, Mujeen Sung, Jaewoo Kang, and Danqi Chen. 2021. Learning Dense Representations of Phrases at Scale. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6634–6647, Online. Association for Computational Linguistics.

Jinhyuk Lee, Seongjun Yun, Hyunjae Kim, Miyoung Ko, and Jaewoo Kang. 2018a. Ranking Paragraphs for Improving Answer Recall in Open-Domain Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 565–569, Brussels, Belgium. Association for Computational Linguistics.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019a. Latent Retrieval for Weakly Supervised Open Domain Question Answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.

Kyungjae Lee, Sunghyun Park, Hojae Han, Jinyoung Yeo, Seung-won Hwang, and Juho Lee. 2019b. Learning with Limited Data for Multilingual Reading Comprehension. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2833–2843, Hong Kong, China. Association for Computational Linguistics.

Kyungjae Lee, Kyoungho Yoon, Sunghyun Park, and Seung-won Hwang. 2018b. Semi-supervised Training Data Generation for Multilingual Question Answering. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Sabine Lehmann, Stephan Oepen, Sylvie Regnier-Prost, Klaus Netter, Veronika Lux, Judith Klein, Kirsten Falkedal, Frederik Fouvry, Dominique Estival, Eva Dauphin, Herve Compagnion, Judith Baur, Lorna Balkan, and Doug Arnold. 1996. TSNLP - Test Suites for Natural Language Processing. In *COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics*.

Wendy Grace Lehnert. 1977. *The process of question answering*. PhD Thesis, Yale University.

Tao Lei, Hrishikesh Joshi, Regina Barzilay, Tommi Jaakkola, Kateryna Tymoshenko, Alessandro Moschitti, and Lluís Màrquez. 2016. Semi-supervised Question Retrieval with Gated Convolutions. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1279–1289, San Diego, California. Association for Computational Linguistics.

Gennadi Lembersky, Noam Ordan, and Shuly Wintner. 2011. Language Models for Machine Translation: Original vs. Translated Texts. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 363–374, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. *CoRR*, abs/2104.08691. ArXiv: 2104.08691.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-Shot Relation Extraction via Reading Comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342, Vancouver, Canada. Association for Computational Linguistics.

David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. 2004. RCV1: A New Benchmark Collection for Text Categorization Research. *J. Mach. Learn. Res.*, 5:361–397. Publisher: JMLR.org.

Mike Lewis and Angela Fan. 2018. Generative Question Answering: Learning to Answer the Whole Question. In *International Conference on Learning Representations*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Patrick Lewis, Ludovic Denoyer, and Sebastian Riedel. 2019. Unsupervised Question Answering by Cloze Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4896–4910, Florence, Italy. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020b. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. 2021. Question and Answer Test-Train Overlap in Open-Domain Question Answering Datasets. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1000–1008, Online. Association for Computational Linguistics.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A Diversity-Promoting Objective Function for Neural Conversation Models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

Margaret Li, Jason Weston, and Stephen Roller. 2019. ACUTE-EVAL: Improved Dialogue Evaluation with Optimized Questions and Multi-turn Comparisons. *ArXiv*, abs/1909.03087.

Xiang Lisa Li and Percy Liang. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.

Xin Li and Dan Roth. 2002. Learning Question Classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Zhongli Li, Wenhui Wang, Li Dong, Furu Wei, and Ke Xu. 2020. Harvesting and Refining Question-Answer Pairs for Unsupervised QA. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6719–6728, Online. Association for Computational Linguistics.

Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, Xiaodong Fan, Ruofei Zhang, Rahul Agrawal, Edward Cui, Sining Wei, Taroon Bharti, Ying Qiao, Jiun-Hung Chen, Winnie Wu, Shuguang Liu, Fan Yang, Daniel Campos, Rangan Majumder, and Ming Zhou. 2020. XGLUE: A New Benchmark Dataset for Cross-lingual Pre-training, Understanding and Generation. *arXiv:2004.01401 [cs]*. ArXiv: 2004.01401.

Seungyoung Lim, Myungji Kim, and Jooyoul Lee. 2019. KorQuAD1.0: Korean QA Dataset for Machine Reading Comprehension. *arXiv:1909.07005v2 [cs.CL]*.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yankai Lin, Haozhe Ji, Zhiyuan Liu, and Maosong Sun. 2018. Denoising Distantly Supervised Open-Domain Question Answering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1736–1745, Melbourne, Australia. Association for Computational Linguistics.

Allison Linn. 2018. Microsoft creates AI that can read a document and answer questions about it as well as a person.

Jiahua Liu, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2019a. XQA: A Cross-lingual Open-domain Question Answering Dataset. In *Proceedings of ACL 2019*.

Linqing Liu, Patrick Lewis, Sebastian Riedel, and Pontus Stenetorp. 2021a. Challenges in Generalization in Open Domain Question Answering. *arXiv:2109.01156 [cs]*. ArXiv: 2109.01156.

Pengyuan Liu, Yuning Deng, Chenghao Zhu, and Han Hu. 2019b. XCMRC: Evaluating Cross-lingual Machine Reading Comprehension. *arXiv:1908.05416 [cs]*. ArXiv: 1908.05416.

Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating Wikipedia by Summarizing Long Sequences. In *International Conference on Learning Representations*.

Shangqing Liu, Yu Chen, Xiaofei Xie, Jing Kai Siow, and Yang Liu. 2021b. Retrieval-Augmented Generation for Code Summarization via Hybrid GNN. In *International Conference on Learning Representations*.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021c. GPT Understands, Too. *CoRR*, abs/2103.10385. ArXiv: 2103.10385.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019c. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv:1907.11692 [cs]*. ArXiv: 1907.11692.

Robert L. Logan IV, Ivana Balazevic, Eric Wallace, Fabio Petroni, Sameer Singh, and Sebastian Riedel. 2021. Cutting Down on Prompts and Parameters: Simple Few-Shot Learning with Language Models. *CoRR*, abs/2106.13353. ArXiv: 2106.13353.

Shayne Longpre, Yi Lu, and Joachim Daiber. 2021. MKQA: A Linguistically Diverse Benchmark for Multilingual Open Domain Question Answering. *Transactions of the Association for Computational Linguistics*, 9:1389–1406. _eprint: https://direct.mit.edu/tacl/article-pdf/doi/10.1162/tacl_a_00433/1976187/tacl_a_00433.pdf.

George F. Luger. 2008. *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*, 6th edition. Addison-Wesley Publishing Company, USA.

H. P. Luhn. 1957. A Statistical Approach to Mechanized Encoding and Searching of Literary Information. *IBM Journal of Research and Development*, 1(4):309–317.

Yu A. Malkov and D. A. Yashunin. 2020. Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):824–836.

Varun Manjunatha, Nirat Saini, and Larry S. Davis. 2019. Explicit Bias Discovery in Visual Question Answering Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2021. Generation-Augmented Retrieval for Open-Domain Question Answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4089–4100, Online. Association for Computational Linguistics.

Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The Penn Treebank: annotating predicate argument structure. In *Proceedings of the workshop on Human Language Technology - HLT '94*, page 114, Plainsboro, NJ. Association for Computational Linguistics.

M. E. Maron and J. L. Kuhns. 1960. On Relevance, Probabilistic Indexing and Information Retrieval. *J. ACM*, 7(3):216–244. Place: New York, NY, USA Publisher: Association for Computing Machinery.

Martı́n Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing

Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems.

Rebecca Marvin and Tal Linzen. 2018. Targeted Syntactic Evaluation of Language Models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

Luca Massarelli, Fabio Petroni, Aleksandra Piktus, Myle Ott, Tim Rocktäschel, Vassilis Plachouras, Fabrizio Silvestri, and Sebastian Riedel. 2020. How Decoding Strategies Affect the Verifiability of Generated Text. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 223–235, Online. Association for Computational Linguistics.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6297–6308.

Bryan McCann, N. Keskar, Caiming Xiong, and R. Socher. 2018. The Natural Language Decathlon: Multitask Learning as Question Answering. *ArXiv*, abs/1806.08730.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. 2018. Mixed Precision Training. In *ICLR*.

Todor Mihaylov and Anette Frank. 2018. Knowledgeable Reader: Enhancing Cloze-Style Reading Comprehension with External Commonsense Knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational*

Linguistics (Volume 1: Long Papers), pages 821–832, Melbourne, Australia. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and Their Compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119, USA. Curran Associates Inc. Event-place: Lake Tahoe, Nevada.

Sewon Min, Jordan Boyd-Graber, Chris Alberti, Danqi Chen, Eunsol Choi, Michael Collins, Kelvin Guu, Hannaneh Hajishirzi, Kenton Lee, Jennimaria Palomaki, Colin Raffel, Adam Roberts, Tom Kwiatkowski, Patrick Lewis, Yuxiang Wu, Heinrich Küttler, Linqing Liu, Pasquale Minervini, Pontus Stenetorp, Sebastian Riedel, Sohee Yang, Minjoon Seo, Gautier Izacard, Fabio Petroni, Lucas Hosseini, Nicola De Cao, Edouard Grave, Ikuya Yamada, Sonse Shimaoka, Masatoshi Suzuki, Shumpei Miyawaki, Shun Sato, Ryo Takahashi, Jun Suzuki, Martin Fajcik, Martin Docekal, Karel Ondrej, Pavel Smrz, Hao Cheng, Yelong Shen, Xiaodong Liu, Pengcheng He, Weizhu Chen, Jianfeng Gao, Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Wen-tau Yih. 2021. NeurIPS 2020 EfficientQA Competition: Systems, Analyses and Lessons Learned. In *Proceedings of the NeurIPS 2020 Competition and Demonstration Track*, volume 133 of *Proceedings of Machine Learning Research*, pages 86–111. PMLR.

Sewon Min, Danqi Chen, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019a. A Discrete Hard EM Approach for Weakly Supervised Question Answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2851–2864, Hong Kong, China. Association for Computational Linguistics.

Sewon Min, Danqi Chen, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019b.

Knowledge Guided Text Retrieval and Reading for Open Domain Question Answering. *CoRR*, abs/1911.03868. ArXiv: 1911.03868.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering Ambiguous Open-domain Questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797, Online. Association for Computational Linguistics.

Sewon Min, Victor Zhong, Richard Socher, and Caiming Xiong. 2018. Efficient and Robust Question Answering from Minimal Context over Documents. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1725–1735, Melbourne, Australia. Association for Computational Linguistics.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, Suntec, Singapore. Association for Computational Linguistics.

Tom M. Mitchell, Justin Betteridge, Andrew Carlson, Estevam Hruschka, and Richard Wang. 2009. Populating the Semantic Web by Macro-reading Internet Text. In *The Semantic Web - ISWC 2009*, pages 998–1002, Berlin, Heidelberg. Springer Berlin Heidelberg.

Bhaskar Mitra, Fernando Diaz, and Nick Craswell. 2017. Learning to Match Using Local and Distributed Representations of Text for Web Search. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, pages 1291–1299, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee. Event-place: Perth, Australia.

Nikita Moghe, Siddhartha Arora, Suman Banerjee, and Mitesh M. Khapra. 2018. Towards Exploiting Background Knowledge for Building Conversation Systems.

In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2322–2332, Brussels, Belgium. Association for Computational Linguistics.

Dan Moldovan, Marius Paşca, Sanda Harabagiu, and Mihai Surdeanu. 2003. Performance issues and error analysis in an open-domain question answering system. *ACM Transactions on Information Systems*, 21(2):133–154.

Hussein Mozannar, Elie Maamary, Karl El Hajal, and Hazem Hajj. 2019. Neural Arabic Question Answering. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 108–118, Florence, Italy. Association for Computational Linguistics.

Stephen Mussmann and Stefano Ermon. 2016. Learning and Inference via Maximum Inner Product Search. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2587–2596, New York, New York, USA. PMLR.

Timo Möller, Julian Risch, and Malte Pietsch. 2021. GermanQuAD and GermanDPR: Improving Non-English Question Answering and Passage Retrieval. In *Proceedings of the 3rd Workshop on Machine Reading for Question Answering*, pages 42–50, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Preksha Nema and Mitesh M. Khapra. 2018. Towards a Better Metric for Evaluating Question Generation Systems. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3950–3959, Brussels, Belgium. Association for Computational Linguistics.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *CoRR*, abs/1901.04085. ArXiv: 1901.04085.

Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document Expansion by Query Prediction. *arXiv:1904.08375 [cs]*. ArXiv: 1904.08375.

Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Sejr Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Scott Yih. 2020. Unified Open-Domain Question Answering with Structured and Unstructured Knowledge. *CoRR*, abs/2012.14610. ArXiv: 2012.14610.

Barlas Oguz, Kushal Lakhotia, Anchit Gupta, Patrick S. H. Lewis, Vladimir Karpukhin, Aleksandra Piktus, Xilun Chen, Sebastian Riedel, Wen-tau Yih, Sonal Gupta, and Yashar Mehdad. 2021. Domain-matched Pre-training Tasks for Dense Retrieval. *CoRR*, abs/2107.13602. ArXiv: 2107.13602.

Andrew M. Olney, Arthur C. Graesser, and Natalie K. Person. 2012. Question Generation from Concept Maps. *Dialogue & Discourse*, 3(2):75–99–99.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Lixin Su, and Xueqi Cheng. 2019. HAS-QA: Hierarchical Answer Spans Model for Open-Domain Question Answering. In *AAAI*, pages 6875–6882. AAAI Press.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d\textquotesingle Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Ethan Perez, Siddharth Karamcheti, Rob Fergus, Jason Weston, Douwe Kiela, and Kyunghyun Cho. 2019. Finding Generalizable Evidence by Learning to Convince Q&A Models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2402–2411, Hong Kong, China. Association for Computational Linguistics.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. True Few-Shot Learning with Language Models. In *Thirty-Fifth Conference on Neural Information Processing Systems*.

Ethan Perez, Patrick Lewis, Wen-tau Yih, Kyunghyun Cho, and Douwe Kiela. 2020. Unsupervised Question Decomposition for Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8864–8880, Online. Association for Computational Linguistics.

Charles A. Perfetti, Julie van Dyke, and Lesley Hart. 2001. The psycholinguistics of basic literacy. *Annual Review of Applied Linguistics*, 21:127–149. Publisher: Cambridge University Press.

Carol Peters. 2001. Introduction. In *Cross-Language Information Retrieval and Evaluation*, pages 1–6, Berlin, Heidelberg. Springer Berlin Heidelberg.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2020. How Context Affects Language Models' Factual Predictions. In *Automated Knowledge Base Construction*.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a Benchmark for Knowledge Intensive Language Tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language Models as Knowledge Bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Dmytro Okhonko, Samuel Broscheit, Gautier Izacard, Patrick Lewis, Barlas Oğuz, Edouard Grave, Wen-tau Yih, and Sebastian Riedel. 2022. The Web Is Your Oyster - Knowledge-Intensive NLP against a Very Large Web Corpus. Technical Report arXiv:2112.09924, arXiv. ArXiv:2112.09924 [cs] type: article.

Nina Poerner, Ulli Waltinger, and Hinrich Schütze. 2019. BERT is Not a Knowledge

Base (Yet): Factual Knowledge vs. Name-Based Reasoning in Unsupervised QA. *arXiv preprint arXiv:1911.03681*.

David Poole, Alan Mackworth, and Randy Goebel. 1997. *Computational Intelligence: A Logical Approach*. Oxford University Press, Inc., USA.

John Prager, Eric Brown, Anni Coden, and Dragomir Radev. 2000. Question-Answering by Predictive Annotation. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, pages 184–191, New York, NY, USA. Association for Computing Machinery. Event-place: Athens, Greece.

Raul Puri, Ryan Spring, Mohammad Shoeybi, Mostofa Patwary, and Bryan Catanzaro. 2020. Training Question Answering Models From Synthetic Data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5811–5826, Online. Association for Computational Linguistics.

Guanghui Qin and Jason Eisner. 2021. Learning How to Ask: Querying LMs with Mixtures of Soft Prompts. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5203–5212, Online. Association for Computational Linguistics.

Chen Qu, Liu Yang, Cen Chen, Minghui Qiu, W. Bruce Croft, and Mohit Iyyer. 2020. Open-Retrieval Conversational Question Answering. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, pages 539–548, New York, NY, USA. Association for Computing Machinery. Event-place: Virtual Event, China.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving Language Understanding by Generative Pre-Training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Ori Ram, Yuval Kirstain, Jonathan Berant, Amir Globerson, and Omer Levy. 2021. Few-Shot Question Answering by Pretraining Span Selection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3066–3079, Online. Association for Computational Linguistics.

Parikshit Ram and Alexander G. Gray. 2012. Maximum Inner-Product Search Using Cone Trees. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 931–939, New York, NY, USA. Association for Computing Machinery. Event-place: Beijing, China.

Revanth Gangi Reddy, Bhavani Iyer, Md Arafat Sultan, Rong Zhang, Avi Sil, Vittorio Castelli, Radu Florian, and Salim Roukos. 2020. End-to-End QA on COVID-19: Domain Adaptation with Synthetic Training. *arXiv:2012.01414 [cs]*. ArXiv: 2012.01414.

Revanth Gangi Reddy, Bhavani Iyer, Md Arafat Sultan, Rong Zhang, Avirup Sil, Vittorio Castelli, Radu Florian, and Salim Roukos. 2021a. Synthetic Target Domain Supervision for Open Retrieval QA. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, pages 1793–1797, New York, NY, USA. Association for Computing Machinery. Event-place: Virtual Event, Canada.

Revanth Gangi Reddy, Vikas Yadav, Md Arafat Sultan, Martin Franz, Vittorio Castelli, Heng Ji, and Avirup Sil. 2021b. Towards Robust Neural Retrieval Models with Synthetic Pre-Training. *CoRR*, abs/2104.07800. ArXiv: 2104.07800.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A Conversational Question Answering Challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

David Reinsel, John Gantz, and John Ryding. 2018. The Digitization of the World. White Paper US44413318, IDC.

Ehud Reiter and Robert Dale. 2000. *Building Natural Language Generation Systems*. Cambridge University Press, USA.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond Accuracy: Behavioral Testing of NLP Models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Kyle Richardson, Hai Hu, Lawrence Moss, and Ashish Sabharwal. 2020. Probing Natural Language Inference Models through Semantic Fragments. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8713–8721.

Kyle Richardson and Ashish Sabharwal. 2020. What Does My QA Model Know? Devising Controlled Probes Using Expert Knowledge. *Transactions of the Association for Computational Linguistics*, 8:572–588.

Matthew Richardson. 2013. MCTest: A Challenge Dataset for the Open-Domain

Machine Comprehension of Text. In *Proceedings of the 2013 Conference on Emprical Methods in Natural Language Processing (EMNLP 2013)*.

Ellen Riloff and Michael Thelen. 2000. A Rule-based Question Answering System for Reading Comprehension Tests. In *ANLP-NAACL 2000 Workshop: Reading Comprehension Tests as Evaluation for Computer-Based Language Understanding Systems*.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How Much Knowledge Can You Pack Into the Parameters of a Language Model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

S. E. Robertson and K. Sparck Jones. 1976. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27(3):129–146. _eprint: https://asistdl.onlinelibrary.wiley.com/doi/pdf/10.1002/asi.4630270302.

S.E. Robertson. 1977. The Probability Ranking Principle in IR. *Journal of Documentation*, 33(4):294–304. Publisher: MCB UP Ltd.

Stephen Robertson. 2008. On the history of evaluation in IR. *Journal of Information Science*, 34(4):439–456. _eprint: https://doi.org/10.1177/0165551507086989.

Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.*, 3(4):333–389. Place: Hanover, MA, USA Publisher: Now Publishers Inc.

Pedro Rodriguez and Jordan Boyd-Graber. 2021. Evaluation Paradigms in Question Answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9630–9642, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Pedro Rodriguez, Shi Feng, Mohit Iyyer, He He, and Jordan Boyd-Graber. 2019. Quizbowl: The Case for Incremental Question Answering. *arXiv:1904.04792 [cs]*. ArXiv: 1904.04792.

Anna Rogers, Matt Gardner, and Isabelle Augenstein. 2021. QA Dataset Explosion: A Taxonomy of NLP Resources for Question Answering and Reading Comprehension. *CoRR*, abs/2107.12708. ArXiv: 2107.12708.

BR Rowe, DW Wood, AN Link, and DA Simoni. 2010. Economic impact assessment of NIST's Text REtrieval Conference (TREC) program. *RTI International, July*.

Vasile Rus, Brendan Wyse, Paul Piwek, Mihai Lintean, Svetlana Stoyanchev, and Cristian Moldovan. 2010. The First Question Generation Shared Task Evaluation Challenge. In *Proceedings of the 6th International Natural Language Generation Conference*, INLG '10, pages 251–257, Stroudsburg, PA, USA. Association for Computational Linguistics. Event-place: Trim, Co. Meath, Ireland.

Stuart J. Russell and Peter Norvig. 2003. *Artificial Intelligence: A Modern Approach*, 2 edition. Pearson Education.

Devendra Sachan, Mostofa Patwary, Mohammad Shoeybi, Neel Kant, Wei Ping, William L. Hamilton, and Bryan Catanzaro. 2021. End-to-End Training of Neural Retrievers for Open-Domain Question Answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6648–6662, Online. Association for Computational Linguistics.

Marzieh Saeidi, Max Bartolo, Patrick Lewis, Sameer Singh, Tim Rocktäschel, Mike Sheldon, Guillaume Bouchard, and Sebastian Riedel. 2018. Interpretation of Natural Language Rules in Conversational Machine Reading. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2087–2097, Brussels, Belgium. Association for Computational Linguistics.

G. Salton, A. Wong, and C. S. Yang. 1975. A Vector Space Model for Automatic Indexing. *Commun. ACM*, 18(11):613–620. Place: New York, NY, USA Publisher: Association for Computing Machinery.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523.

Roger C Schank and Robert P Abelson. 1977. *Scripts, plans, goals and understanding: An inquiry into human knowledge structures*. Lawrence Erlbaum.

Timo Schick and Hinrich Schütze. 2021a. Exploiting Cloze-Questions for Few-Shot Text Classification and Natural Language Inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021b. It's Not Just Size That Matters: Small Language Models Are Also Few-Shot Learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online. Association for Computational Linguistics.

Katja Geertruida Schmahl, Tom Julian Viering, Stavros Makrodimitris, Arman Naseri Jahfari, David Tax, and Marco Loog. 2020. Is Wikipedia succeeding in reducing gender bias? Assessing changes in gender bias in Wikipedia using word embeddings. In *Proceedings of the Fourth Workshop on Natural Language Processing and Computational Social Science*, pages 94–103, Online. Association for Computational Linguistics.

Victor Schmidt, Kamal Goyal, Aditya Joshi, Boris Feld, Liam Conell, Nikolas Laskaris, Doug Blank, Jonathan Wilson, Sorelle Friedler, and Sasha Luccioni. 2021. CodeCarbon: Estimate and Track Carbon Emissions from Machine Learning Computing. *Online*. Publisher: Zenodo.

Klaus Schwab. 2019. The Global Competitiveness Report 2019. Technical report, World Economic Forum.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2021. WikiMatrix: Mining 135M Parallel Sentences in 1620 Language

Pairs from Wikipedia. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1351–1361, Online. Association for Computational Linguistics.

Holger Schwenk and Xian Li. 2018. A Corpus for Multilingual Document Classification in Eight Languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, Miyazaki, Japan. European Languages Resources Association (ELRA).

Christopher Sciavolino, Zexuan Zhong, Jinhyuk Lee, and Danqi Chen. 2021. Simple Entity-Centric Questions Challenge Dense Retrievers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6138–6148, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *iclr*.

Minjoon Seo, Tom Kwiatkowski, Ankur Parikh, Ali Farhadi, and Hannaneh Hajishirzi. 2018. Phrase-Indexed Question Answering: A New Challenge for Scalable Document Comprehension. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 559–564, Brussels, Belgium. Association for Computational Linguistics.

Minjoon Seo, Jinhyuk Lee, Tom Kwiatkowski, Ankur Parikh, Ali Farhadi, and Hannaneh Hajishirzi. 2019. Real-Time Open-Domain Question Answering with Dense-Sparse Phrase Index. In *Proceedings of the 57th Annual Meeting of the As-*

sociation for Computational Linguistics, pages 4430–4441, Florence, Italy. Association for Computational Linguistics.

Iulian Vlad Serban, Alberto García-Durán, Caglar Gulcehre, Sungjin Ahn, Sarath Chandar, Aaron Courville, and Yoshua Bengio. 2016. Generating Factoid Questions With Recurrent Neural Networks: The 30M Factoid Question-Answer Corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 588–598, Berlin, Germany. Association for Computational Linguistics.

Chih Chieh Shao, Trois Liu, Yuting Lai, Yiying Tseng, and Sam Tsai. 2018. DRCD: a Chinese Machine Reading Comprehension Dataset. *arXiv:1806.00920 [cs]*. ArXiv: 1806.00920.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, Online. Association for Computational Linguistics.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval Augmentation Reduces Hallucination in Conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chenglei Si, Chen Zhao, and Jordan Boyd-Graber. 2021. What's in a Name? Answer Equivalence For Open-Domain Question Answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9623–9629, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

R. F. Simmons. 1965. Answering English Questions by Computer: A Survey. *Communications of the Association for Computing Machinery*, 8(1):53–70.

Robert F. Simmons, Sheldon Klein, and Keren McConlogue. 1964. Indexing and dependency logic for answering english questions. *American Documentation*, 15:196–204.

Jasdeep Singh, Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2019. XLDA: Cross-Lingual Data Augmentation for Natural Language Inference and Question Answering. *arXiv:1905.11471 [cs]*. ArXiv: 1905.11471.

Amit Singhal. 2001. Modern information retrieval: a brief overview. *BULLETIN OF THE IEEE COMPUTER SOCIETY TECHNICAL COMMITTEE ON DATA ENGINEERING*, 24:2001.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2016. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. *arXiv:1612.03975 [cs]*. ArXiv: 1612.03975.

Mitchell Stern, Jacob Andreas, and Dan Klein. 2017. A Minimal Span-Based Neural Constituency Parser. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 818–827, Vancouver, Canada. Association for Computational Linguistics.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and Policy Considerations for Deep Learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.

Sandeep Subramanian, Guillaume Lample, Eric Michael Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y.-Lan Boureau. 2018. Multiple-Attribute Text Style Transfer. *arXiv:1811.00552 [cs]*. ArXiv: 1811.00552.

Saku Sugawara, Kentaro Inui, Satoshi Sekine, and Akiko Aizawa. 2018. What Makes Reading Comprehension Questions Easier? In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4208–4219, Brussels, Belgium. Association for Computational Linguistics.

Saku Sugawara, Pontus Stenetorp, Kentaro Inui, and Akiko Aizawa. 2020. Assessing the Benchmarking Capacity of Machine Reading Comprehension Datasets. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8918–8927. AAAI Press.

Sainbayar Sukhbaatar, arthur szlam, Jason Weston, and Rob Fergus. 2015. End-To-End Memory Networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2440–2448. Curran Associates, Inc.

M. Surdeanu. 2013. Overview of the TAC2013 Knowledge Base Population Evaluation: English Slot Filling and Temporal Slot Filling. *TAC*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Alon Talmor and Jonathan Berant. 2018. The Web as a Knowledge-Base for Answering Complex Questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 641–651, New Orleans, Louisiana. Association for Computational Linguistics.

Alon Talmor and Jonathan Berant. 2019. MultiQA: An Empirical Investigation of Generalization and Transfer in Reading Comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4911–4921, Florence, Italy. Association for Computational Linguistics.

Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. 2020. oLMpics-On What Language Model Pre-training Captures. *Transactions of the Association for Computational Linguistics*, 8:743–758.

Alon Talmor, Ori Yoran, Amnon Catav, Dan Lahav, Yizhong Wang, Akari Asai, Gabriel Ilharco, Hannaneh Hajishirzi, and Jonathan Berant. 2021. Multi-Modal{QA}: complex question answering over text, tables and images. In *International Conference on Learning Representations*.

Derek Tam, Rakesh R. Menon, Mohit Bansal, Shashank Srivastava, and Colin Raffel. 2021. Improving and Simplifying Pattern Exploiting Training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4980–4991, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. 2021. Long Range Arena : A Benchmark for Efficient Transformers. In *International Conference on Learning Representations*.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2020. Efficient Transformers: A Survey. *CoRR*, abs/2009.06732. ArXiv: 2009.06732.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019a. BERT Rediscovers the Classical NLP Pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. 2019b. What do you learn from context? Probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

James Thorne and Andreas Vlachos. 2021. Elastic weight consolidation for better bias inoculation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 957–964, Online. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a Large-scale Dataset for Fact Extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

JP Thorne. 1962. Automatic language analysis. ASTIA 297381. Technical report, Final Tech Rep, Arlington, Va.

Alan Turing. 1950. Computing Machinery and Intelligence. *Mind*, LIX(236):433–460. _eprint: https://academic.oup.com/mind/article-pdf/LIX/236/433/30123314/lix-236-433.pdf.

Christina Unger, Corina Forescu, Vanessa Lopez, Axel-Cyrille Ngonga Ngomo, Elena Cabrio, Philipp Cimiano, and Sebastian Walter. 2015. Question Answering over Linked Data (QALD-5). In *CLEF*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, \Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 6000–6010, Red Hook, NY, USA. Curran Associates Inc. Event-place: Long Beach, California, USA.

Pat Verga, Haitian Sun, Livio Baldini Soares, and William W. Cohen. 2020. Facts as Experts: Adaptable and Interpretable Neural Memory over Symbolic Knowledge. *arXiv:2007.00849 [cs]*. ArXiv: 2007.00849.

Jesse Vig and Yonatan Belinkov. 2019. Analyzing the Structure of Attention in a Transformer Language Model. *Second BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*.

Ashwin Vijayakumar, Michael Cogswell, Ramprasaath Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2018. Diverse Beam Search for Improved Description of Complex Scenes. *AAAI Conference on Artificial Intelligence*.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and Composing Robust Features with Denoising Autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 1096–1103, New York, NY, USA. Association for Computing Machinery. Event-place: Helsinki, Finland.

Vered Volansky, Noam Ordan, and Shuly Wintner. 2015. On the features of translationese. *Digital Scholarship in the Humanities*, 30(1):98–118.

Ellen M. Voorhees. 2002a. Overview of the TREC 2002 Question Answering Track. In *Proceedings of The Eleventh Text REtrieval Conference, TREC 2002, Gaithersburg, Maryland, USA, November 19-22, 2002*, volume 500-251 of *NIST Special Publication*. National Institute of Standards and Technology (NIST).

Ellen M. Voorhees. 2002b. The Philosophy of Information Retrieval Evaluation. In *Evaluation of Cross-Language Information Retrieval Systems*, pages 355–370, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ellen M. Voorhees and Donna K. Harman, editors. 1999. *Proceedings of The Eighth Text REtrieval Conference, TREC 1999, Gaithersburg, Maryland, USA, November 17-19, 1999*, volume 500-246 of *NIST Special Publication*. National Institute of Standards and Technology (NIST).

Ellen M. Voorhees and Dawn M. Tice. 2000. Building a Question Answering Test Collection. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, pages 200–207, New York, NY, USA. ACM. Event-place: Athens, Greece.

Claudia Wagner, Eduardo Graells-Garrido, David Garcia, and Filippo Menczer. 2016. Women through the glass ceiling: gender asymmetries in Wikipedia. *EPJ Data Science*, 5(1):5.

Eric Wallace, Pedro Rodriguez, Shi Feng, Ikuya Yamada, and Jordan Boyd-Graber. 2019. Trick Me If You Can: Human-in-the-Loop Generation of Adversarial Examples for Question Answering. *Transactions of the Association for Computational Linguistics*, 7:387–401.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d\textquotesingle Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 3261–3275. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018a. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Liang Wang, Sujian Li, Wei Zhao, Kewei Shen, Meng Sun, Ruoyu Jia, and Jingming Liu. 2018b. Multi-Perspective Context Aggregation for Semi-supervised Cloze-style Reading Comprehension. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 857–867, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Shuohang Wang and Jing Jiang. 2017. Machine comprehension using match-LSTM and answer pointer. In *ICLR 2017: International Conference on Learning Representations, Toulon, France, April 24-26: Proceedings*, pages 1–15.

Shuohang Wang, Mo Yu, Xiaoxiao Guo, Zhiguo Wang, Tim Klinger, Wei Zhang, Shiyu Chang, Gerry Tesauro, Bowen Zhou, and Jing Jiang. 2018c.

R\(^\mbox3\): Reinforced Ranker-Reader for Open-Domain Question Answering. In *AAAI*, pages 5981–5988. AAAI Press.

Shuohang Wang, Mo Yu, Jing Jiang, Wei Zhang, Xiaoxiao Guo, Shiyu Chang, Zhiguo Wang, Tim Klinger, Gerald Tesauro, and Murray Campbell. 2018d. Evidence Aggregation for Answer Re-Ranking in Open-Domain Question Answering. In *ICLR (Poster)*. OpenReview.net.

W. Wang, J. Auer, R. Parasuraman, I. Zubarev, D. Brandyberry, and M. P. Harper. 2000. A Question Answering System Developed as a Project in a Natural Language Processing Course. In *ANLP-NAACL 2000 Workshop: Reading Comprehension Tests as Evaluation for Computer-Based Language Understanding Systems*.

Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. 2017. Gated Self-Matching Networks for Reading Comprehension and Question Answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 189–198, Vancouver, Canada. Association for Computational Linguistics.

William Wang, Angelina Wang, Aviv Tamar, Xi Chen, and Pieter Abbeel. 2018e. Safer Classification by Synthesis. *arXiv:1711.08534 [cs, stat]*. ArXiv: 1711.08534.

Alex Warstadt and Samuel R Bowman. 2019. Linguistic Analysis of Pretrained Sentence Encoders with Acceptability Judgments. *arXiv preprint arXiv:1901.03438*.

Alex Warstadt, Yu Cao, Ioana Grosu, Wei Peng, Hagen Blix, Yining Nie, Anna Alsop, Shikha Bordia, Haokun Liu, Alicia Parrish, Sheng-Fu Wang, Jason Phang, Anhad Mohananey, Phu Mon Htut, Paloma Jeretic, and Samuel R. Bowman. 2019. Investigating BERT's Knowledge of Language: Five Analysis Methods with NPIs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural*

Language Processing (EMNLP-IJCNLP), pages 2877–2887, Hong Kong, China. Association for Computational Linguistics.

Dirk Weissenborn, Tomáš Kočiský, and Chris Dyer. 2017a. Dynamic Integration of Background Knowledge in Neural NLU Systems. *arXiv:1706.02596 [cs]*. ArXiv: 1706.02596.

Dirk Weissenborn, Georg Wiese, and Laura Seiffe. 2017b. Making Neural QA as Simple as Possible but not Simpler. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 271–280, Vancouver, Canada. Association for Computational Linguistics.

Jason Weston, Sumit Chopra, and Antoine Bordes. 2015. Memory Networks. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Jason Weston, Emily Dinan, and Alexander Miller. 2018. Retrieve and Refine: Improved Sequence Generation Models For Dialogue. In *Proceedings of the 2018 EMNLP Workshop SCAI: The 2nd International Workshop on Search-Oriented Conversational AI*, pages 87–92, Brussels, Belgium. Association for Computational Linguistics.

Georg Wiese, Dirk Weissenborn, and Mariana Neves. 2017. Neural Domain Adaptation for Biomedical Question Answering. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 281–289, Vancouver, Canada. Association for Computational Linguistics.

The WikiMedia Foundation. 2021. Wikipedia:Copyrights. Page Version ID: 1035576274.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Wenhan Xiong, Mo Yu, Xiaoxiao Guo, Hong Wang, Shiyu Chang, Murray Campbell, and William Yang Wang. 2019. Simple yet Effective Bridge Reasoning for Open-Domain Multi-Hop Question Answering. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 48–52, Hong Kong, China. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Vikas Yadav, Steven Bethard, and Mihai Surdeanu. 2019. Alignment over Heterogeneous Embeddings for Question Answering. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2681–2691, Minneapolis, Minnesota. Association for Computational Linguistics.

Wei Yang, Yuqing Xie, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019a. Data Augmentation for BERT Fine-Tuning in Open-Domain Question Answering. *arXiv:1904.06652 [cs]*. ArXiv: 1904.06652.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019b. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.

Zhilin Yang, Junjie Hu, Ruslan Salakhutdinov, and William Cohen. 2017. Semi-Supervised QA with Generative Domain-Adaptive Nets. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1040–1050, Vancouver, Canada. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Xuchen Yao, Gosse Bouma, and Yi Zhang. 2012. Semantics-based Question Generation and Implementation. *Dialogue & Discourse*, 3(2):11–42.

Qinyuan Ye, Madian Khabsa, Mike Lewis, Sinong Wang, Xiang Ren, and Aaron Jaech. 2021. Sparse Distillation: Speeding Up Text Classification by Using Bigger Models. *arXiv:2110.08536 [cs]*. ArXiv: 2110.08536.

Wen-tau Yih, Kristina Toutanova, John C. Platt, and Christopher Meek. 2011. Learning Discriminative Projections for Text Similarity Measures. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, CoNLL '11, pages 247–256, USA. Association for Computational Linguistics. Event-place: Portland, Oregon.

Dani Yogatama, Cyprien de Masson d'Autume, Jerome Connor, Tomas Kocisky, Mike Chrzanowski, Lingpeng Kong, Angeliki Lazaridou, Wang Ling, Lei Yu, Chris Dyer, and Phil Blunsom. 2019. Learning and Evaluating General Linguistic Intelligence. *arXiv:1901.11373 [cs, stat]*. ArXiv: 1901.11373.

Adams Wei Yu, David Dohan, Quoc Le, Thang Luong, Rui Zhao, and Kai Chen. 2018. Fast and Accurate Reading Comprehension by Combining Self-Attention and Convolution. In *International Conference on Learning Representations*.

Xingdi Yuan, Tong Wang, Caglar Gulcehre, Alessandro Sordoni, Philip Bachman, Saizheng Zhang, Sandeep Subramanian, and Adam Trischler. 2017. Machine Comprehension by Text-to-Text Neural Question Generation. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 15–25, Vancouver, Canada. Association for Computational Linguistics.

Daniel Zhang, Saurabh Mishra, Erik Brynjolfsson, John Etchemendy, Deep Ganguli, Barbara Grosz, Terah Lyons, James Manyika, Juan Niebles, Michael Sellitto, Yoav Shoham, Jack Clark, and Raymond Perrault. 2021. *The AI Index 2021 Annual Report*. Stanford University.

Shiyue Zhang and Mohit Bansal. 2019. Addressing Semantic Drift in Question Generation for Semi-Supervised Question Answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2495–2509, Hong Kong, China. Association for Computational Linguistics.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2017. Men Also Like Shopping: Reducing Gender Bias Amplification using Corpus-level Constraints. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2979–2989, Copenhagen, Denmark. Association for Computational Linguistics.

Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke. 2018. Paragraph-level Neural Question Generation with Maxout Pointer and Gated Self-attention Networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3901–3910, Brussels, Belgium. Association for Computational Linguistics.

Wanjun Zhong, Jingjing Xu, Duyu Tang, Zenan Xu, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2020. Reasoning Over Semantic-Level Graph for Fact Checking. In *Proceedings of the 58th Annual Meeting of the Association for Com-*

putational Linguistics, pages 6170–6180, Online. Association for Computational Linguistics.

Zexuan Zhong, Dan Friedman, and Danqi Chen. 2021. Factual Probing Is [MASK]: Learning vs. Learning to Recall. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5017–5033, Online. Association for Computational Linguistics.

Pengfei Zhu, Xiaoguang Li, Jian Li, and Hai Zhao. 2021. Unsupervised Open-Domain Question Answering. *arXiv:2108.13817 [cs]*. ArXiv: 2108.13817.

Elizaveta Zimina, Jyrki Nummenmaa, Kalervo Jarvelin, Jaakko Peltonen, and Kostas Stefanidis. 2018. MuG-QA: Multilingual Grammatical Question Answering for RDF Data. *2018 IEEE International Conference on Progress in Informatics and Computing (PIC)*, pages 57–61.

Justin Zobel, Alistair Moffat, and Kotagiri Ramamohanarao. 1998. Inverted Files versus Signature Files for Text Indexing. *ACM Trans. Database Syst.*, 23(4):453–490. Place: New York, NY, USA Publisher: Association for Computing Machinery.