Training Datasets for Machine Reading Comprehension and Their Limitations

Johannes Welbl

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 (UCL)

September 5, 2020

I, Johannes Welbl, confirm that the work presented in this report is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

Neural networks are a powerful model class to learn machine Reading Comprehension (RC), yet they crucially depend on the availability of suitable training datasets. In this thesis we describe methods for data collection, evaluate the performance of established models, and examine a number of model behaviours and dataset limitations.

We first describe the creation of a data resource for the science exam QA domain, and compare existing models on the resulting dataset. The collected questions are plausible – non-experts can distinguish them from real exam questions with 55% accuracy – and using them as additional training data leads to improved model scores on real science exam questions.

Second, we describe and apply a distant supervision dataset construction method for multi-hop RC across documents. We identify and mitigate several dataset assembly pitfalls – a lack of unanswerable candidates, label imbalance, and spurious correlations between documents and particular candidates – which often leave shallow predictive cues for the answer. Furthermore we demonstrate that selecting relevant document combinations is a critical performance bottleneck on the datasets created. We thus investigate Pseudo-Relevance Feedback, which leads to improvements compared to TF-IDF-based document combination selection both in retrieval metrics and answer accuracy.

Third, we investigate model undersensitivity: model predictions do not change when given adversarially altered questions in SQUAD2.0 and NEWSQA, even though they should. We characterise affected samples, and show that the phenomenon is related to a lack of structurally similar but unanswerable samples during training: data augmentation reduces the adversarial error rate, e.g. from 51.7% to 20.7% for a BERT model on SQuAD2.0, and improves robustness also in other settings. Finally we explore efficient formal model verification via Interval Bound Propagation (IBP) to measure and address model undersensitivity, and show that using an IBP-derived auxiliary loss can improve verification rates, e.g. from 2.8% to 18.4% on the SNLI test set.

Impact Statement

This PhD thesis presents research in the intersection of Natural Language Processing (NLP) and Machine Learning (ML), and its aim is to further the development of computer models for text understanding with a particular focus on the datasets models are trained with.

Systems with intelligent text processing capabilities become increasingly important as the amount of digital textual information expands. They can be used to support querying, structuring, filtering, combining, organising and validating information from text automatically, and thus facilitate access to knowledge in the digital sphere. Concrete example applications include automatic Question Answering (QA) sytems, improved web search, and automatic fact checking. Language understanding forms a core component of artificial intelligence more generally, and improving machine reading comprehension can be expected to enable new applications that integrate it with other sub-fields or input modalities, e.g. vision.

In this thesis we address a set of machine reading comprehension scenarios and study both data annotation methods, the resulting datasets, and the consequent behaviours and limitations of models trained on these datasets. Our observations specifically on different dataset biases can help improve our understanding of machine reading comprehension systems, which is relevant in particular as the overall interpretability of contemporary neural methods is limited. The datasets whose construction is described have been made available for further research to the wider community, and some of the research has led to publication at leading venues in ML and NLP.

Acknowledgements

I am deeply thankful to both Sebastian and Pontus for their guidance and inspiration throughout the journey of this PhD programme, for their patience and continuous support, and for the many hours spent in discussion, pair-writing, and pair-coding – I absolutely couldn't think of any better advisors. Thank you for the tricks, the time, and the trust.

Next I want to thank my examiners Jonathan Berant and Emine Yilmaz for their in-depth engagement and discussion of this thesis, as well as Guillaume Bouchard, David Barber, John Shawe-Taylor and Tim Rocktäschel for direction and feedback along the way.

I extend my gratitude to all past and present members of the UCLMR / UCLNLP group, as well as my internship colleagues at the Allen Institute for Artificial Intelligence and at DeepMind. I consider myself lucky to have found such great research environments to learn in.

I had the pleasure to work with fantastic collaborators during the last years, who have brought me much inspiration, useful feedback and criticism, who have often been role models, and from whom I have learned a lot: thank you Isabelle Augenstein, Max Bartolo, Gérard Biau, Matko Bošnjak, Peter Clark, Michal Daniluk, Sumanth Dathathri, Thomas Demeester, Tim Dettmers, Krishnamurthy (Dj) Dvijotham, Chris Dyer, Richard Evans, Matt Gardner, Sven Gowal, Po-Sen Huang, Pushmeet Kohli, Nelson F. Liu, Pasquale Minervini, Jeff Mitchell, Jason Naradowsky, Alistair Roberts, Marco Concetto Rudilosso, Erwan Scornet, Robert Stanforth, Martin Szummer, Théo Trouillon, Andreas Vlachos, Dirk Weissenborn, Yuxiang Wu, Dani Yogatama, and Takuma Yoneda. I am furthermore grateful to my

previous advisors Fred Hamprecht and Ullrich Köthe for teaching me and preparing me for research.

I would like to thank Patrick Lewis, Maximilian Mozes, Sebastian Ruder and Pasquale Minervini for concrete feedback on parts of this work at different stages. Many thanks also to the Engineering and Physical Sciences Research Council (EPSRC) and UCL Computer Science Department for supporting me with a studentship, which gave me the time and freedom to pursue this research.

The PhD wouldn't have been fun without a fun lab: thanks Tim, Matko, Georgios, Marzieh, Ivan, Isabelle, Tim, Tom, Jeff, Pasquale, Patrick, Yuxiang, Max (and many others!) — without you I wouldn't have enjoyed the journey even half as much. I would probably not have been tempted to embark on ambitious burrito bets about experimental outcomes. Thank you also to Bert and Roberta, it has been fun working with you.

Finally I want to thank Joscha, Annabell, Sven, Christian, Philipp, Karina, Michael and Slava for your friendship and support; and to my hamster for the excellent companionship. Thank you Roisin for showing me the climbing wall. Thanks to Finsi House for being my home through much of these years in London, and in particular to Lara, Alex and Pedro for keeping me sane and happy during the brief lockdown interval in the end. Schliesslich will ich meinen Eltern Werner and Birgit, sowie Katharina and Viky für ihre Liebe und Unterstützung danken. Wenns nix werd gehma halt zum Griech.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The written word plays a fundamental role in organising information: it serves to record memories, store facts and observations, and to share experiences and preserve lessons learnt from them. Reading Comprehension (RC) – the ability to understand the written word – is a learned cognitive capacity that constitutes a central prerequisite for knowledge acquisition from text, and it allows humans to tap into a vast store of collectively assembled historic and contemporary information, e.g. in encyclopediae or the web.

While RC is originally a human skill, the availability of text in digital form, and the need to organise, channel, filter and search in it, raise a both philosophically intriguing, and practically challenging question: to what extent can the ability to understand natural language text be automated in a machine? Research in Natural Language Processing (NLP), and more specifically in machine reading comprehension, aims to answer this question, and in a constructive way: by building systems that demonstrate text comprehension ability (to varying degrees). If a computer system is given a natural language text and probed with a comprehension question about its content – can it produce the correct answer?

Figure 1.1 shows an example from the Stanford Question Answering Dataset (SQuAD; Rajpurkar et al. (2016)), a widely used benchmark dataset for this NLP task, which contains snippets of Wikipedia text together with crowd-sourced questions about their content, and correct answers.

This example illustrates that natural language expressions codify messages in

> […] Robert Guiscard, another Norman adventurer previously elevated to the dignity of count of Apulia as the result of his military successes, ultimately drove the Byzantines out of southern Italy. Having obtained the consent of pope Gregory VII and acting as his vassal, Robert continued his campaign conquering the Balkan peninsula as a foothold for western feudal lords and the Catholic Church. […]
>
> **Q:** What was the name of the count of Apulia?
> **A:** Robert Guiscard

Figure 1.1: Example from the SQuAD1.1 dataset for machine Reading Comprehension.

ways that do not strictly adhere to an easily specifiable set of rules that would allow for programmatic extraction of the content: while the question asks about the "*name of the count of Apulia*", this is not stated in the given text verbatim. Instead, the paragraph describes that this person was "*previously elevated to the dignity of count of Apulia*", which entails being count of Apulia. There are also various alternative ways to formulate an information request about "*the name*" of a person – each with a subtly different meaning: "*Who was ...*", "*Which person was ...*", or "*Which Norman adventurer was ...*". Note that the last formulation is specific to this given context: it includes another property (*"Norman adventurer"*) which would not be found in other contexts in which "*the name*" of an individual is to be identified.

This example illustrates that there is a vast amount of lexical and syntactic diversity in natural language – even to express very similar messages or information requests. The wide range of these variations, which may or may not depend on the context, makes it infeasible to specify a comprehensive set of regular expressions both general and precise enough to capture the underlying pattern in information requests like the above. Interpretations are generally context-sensitive, and various degrees of inference allow a competent reader to go beyond what is literally stated. This inherent variety, redundance, flexibility, and context-sensitivity of natural language renders RC a very challenging task, and for many decades progress within NLP was confined to various subtasks, such as extracting individual entities, and classifying types of relations between them.

But the previous years have established a new paradigm, not only for RC, but for NLP in general: the usage of end-to-end trainable neural networks, which fit continuously parameterised "neural" transformation functions on ideally vast collections of inputs paired with desirable model outputs. For example, the above mentioned SQuAD dataset contains more than 100,000 input-output pairs of comprehension questions and answers obtained via crowdsourcing, without further annotations of entities or other potential intermediate stages of a comprehension process.

The availability of such large training datasets and ready-to-use neural modeling toolkits have spurred a profusion in model and architecture development, and subsequent rapid progress on benchmark datasets, as can be witnessed on the model leaderboard for SQuAD.[1] This line of research has demonstrated that neural models possess sufficient representational capacity to acquire relevant reading skills by fitting correct answering behaviour on comprehension questions end-to-end, and that this generalises to held-out evaluation questions.

The now well-established paradigm for approaching a new RC task has thus become, first, to assemble an ideally large scale dataset of relevant texts and comprehension questions, and second, to design a neural architecture and train it on the assembled data, ideally leveraging pre-trained model weights. This framework is very adaptable, making it possible to interchange different types of datasets, neural architectures, and types of pretrained model weights. It is also a comparatively recently developed paradigm, which opens a variety of research questions about how statistical patterns in the collected training data affect the behaviour of the models trained on it.

On the one hand, it is a core underlying axiom of using a data-driven learning approach that generalisable reading comprehension skills will emerge from statistical patterns when considering a sufficient amount of relevant training examples. And indeed, the "super-human" generalisation scores of neural models on datasets such as SQuAD give justification to this assumption, and raise hopes that the creation of datasets for other RC tasks can lead to similar outcomes. On the other

[1] `https://rajpurkar.github.io/SQuAD-explorer/`

hand, any approach to creating a dataset with samples aimed at demonstrating the necessary skills required for proficiency in a task may come with a misalignment to this objective. It is currently not well understood how precisely the nature of a dataset and its statistical patterns and properties reflect on the skills that a model learns from it, and this extends in particular to RC.

Thus – as datasets form a critical ingredient to build RC systems – both methods for data annotation and the resulting datasets are worth investigation. We will in this thesis describe the creation of several datasets for new RC problems, relate their properties to limitations in RC models trained on them, and evaluate technical solutions to circumvent the identified issues. The thesis will be structured into three main parts which focus on more narrow topics within this general problem setting; we will subsequently give an overview of each.

1.1 Research Overview and Contribution Summary

1.1.1 Dataset Assembly for Reading Comprehension in Science Exams (Part I)

When considering the field of Artificial Intelligence (AI) more broadly, one long-standing challenge has been the evaluation of algorithms on tests designed for probing human intellectual ability and educational progress. A concrete benchmark problem proposed by the Allen Institute for Artificial Intelligence is to evaluate computers on 4^{th} or 8^{th} grade exam questions from natural science subjects. Solving these exams requires a variety of cognitive abilities – including Reading Comprehension – as well as various types of background knowledge and other skills (Jansen et al., 2016). System performance on these exams can thus serve as an aggregate measuring stick for how far the AI field has progressed in building and integrating these capacities into a single computer system.

Reading Comprehension in this context is particularly challenging: the domain is very specific, and the amount of exam questions potentially available for training is limited. But since current neural RC systems rely on ample training data as a critical ingredient, we hypothesise that the assembly of such a data resource can be of use to train RC models and improve their ability to solve science exam questions.

In our first study, we thus develop a data acquisition pipeline with which we can obtain new RC training data specifically for the science exam domain. Based on a large collection of in-domain text and a small collection of real seed exam questions, we collect more than 10,000 questions about study materials in a crowdsourcing task, guided by suggestions of an auxiliary predictor to facilitate annotation. We then compare existing automatic science exam solvers, as well as neural RC models on the resulting dataset, and use the outcome to highlight a number of important dataset properties and limitations. Finally, we demonstrate that the created dataset can be leveraged to improve neural RC system performance on a held out set of real science exam questions.

Summary of Contributions:

1. We describe a method for crowdsourcing RC data in the particular domain of multiple-choice science questions and use it to assemble a new training and evaluation datset resource.

2. We compare the performance of several established science exam solvers and RC models on this new dataset, discuss its properties, and show that it can be used to improve RC performance on real exam questions.

1.1.2 Multi-Hop Reading Comprehension (Part II)

The SQUAD-like dataset construction approach, which we also adopt in Part I, centers around crowdsourcing comprehension questions about a *single* paragraph. The necessary information to answer such a question is very locally concentrated (Min et al., 2018), and there often exists substantial lexical overlap between comprehension question and given document. This naturally limits the scope of the resulting models, as they can overly focus on learning to align and (soft-) match questions with relevant sentences (Weissenborn et al., 2017) which can also be adversarially exploited (Jia and Liang, 2017). Yet text comprehension should ideally go beyond searching for requested information in a single, short piece of text which closely rewords the comprehension question. A more ambitious goal for an RC system is to process several distinct pieces of text, and infer new information from them together. Such “multi-hop” comprehension is a challenging RC task, but even constructing the necessary training and evaluation resources presents a non-trivial and new challenge.

In Part II of this thesis we target this multi-hop comprehension problem, and first assemble dataset resources for an RC scenario where evidence for the correct answer can be combined from multiple documents. We describe the construction of such a data resource in detail, focus on a variety of potential pitfalls and undesirable statistical cues in the process, and investigate ways to overcome and remove these. We then compare several models – statistical, neural, and retrieval-based –, investigate how they reflect data biases, and observe that after removing several salient statistical cues, neural models tend to generalise most robustly to held out samples.

As we observe that one critical aspect to solving multi-document comprehension problems is the identification of relevant text *combinations*, we investigate this selection problem further. Using standard information retrieval (IR) strategies – such as BM25 and TF-IDF – to search for relevant documents has limitations when it comes to retrieving documents whose relevance to a query is co-dependent. A different retrieval method which uses pseudo-relevance feedback (PRF) for query expansion, however, includes content of initially retrieved documents into the search query vector. We examine the suitability of this approach for retrieving multiple related documents, compared to approaches which consider the relevance of documents independently. Finally we combine the retrieval component with a neural RC model, and find that PRF-based systems outperform TF-IDF-based systems on the previously created multi-hop dataset.

Summary of Contributions:

1. We propose a new cross-document multi-hop RC task, and describe a general dataset induction strategy for this proposed problem.

2. We assemble two datasets from different domains for this task, and identify dataset construction pitfalls and remedies.

3. We establish multiple baselines on the resulting multi-step RC data, and analyse model behaviour in detail through ablation studies.

4. We investigate different IR methods for a cross-document RC scenario, in particular Pseudo Relevance Feedback and TF-IDF.

5. We augment established RC methods with multi-step retrieval, and show that jointly ranking document combinations can improve over methods that rank documents independently.

1.1.3 Investigating Model Undersensitivity (Part III)

When neural networks are discriminatively trained to predict correct answers to comprehension questions, they can learn to use arbitrary predictive cues that help them identify the answer. Consequently, it is potentially not necessary for a model to learn adequate representations of comprehension questions and texts in the reconstructive sense: models can instead learn to form predictions based on shallow cues, such as answer type consistency, while disregarding important information in the question.

To substantiate this suspicion we describe an adversarial attack designed to probe an RC model's sensitivity (or lack thereof) to meaning-altering perturbations of comprehension questions which render them unanswerable. Indeed, models exhibit a striking lack of sensitivity to such adversarially chosen question changes, even though exposed to unanswerable questions in the training data. We characterise the affected samples and proceed to investigate modified training strategies as defences against the problem. Based on the results of these experiments we conclude that a lack of structurally similar unanswerable questions in the training set can largely be seen as responsible for problematic model undersensitivity behaviour. Adding such samples into the training set leads to an encouraging yet limited finding: while the cost associated with identifing adversarial samples is substantially increased, the presence of such attacks is not ruled out, and the underlying problem still persists.

We thus study a scenario in which we aim at a more ambitious outcome: we impose a much stronger requirement onto the model and specify that besides fitting the training data, it should *verifiably* not be prone to violate a specification addressing the model's undersensitivity behaviour, which is defined over a combinatorially large space of possible input alterations. Formally verifying neural network behaviour on such large input spaces is a very challenging task with little precedence in NLP. To understand the feasibility and problems associated with this approach we investigate the related, but more focused language understanding task of Natural Language Inference (NLI), which operates on pairs of single sentences. To check whether the undersensitivity specification can be verified, and to train models to become verifiable in this regard, we adopt Interval Bound Propagation, an incomplete method for formal neural network verification. We show that this method is drastically more efficient than exhaustive verification in establishing specification adherence of the Decomposable Attention Model (Parikh et al., 2016), and that when adding a dedicated auxiliary training objective, models can learn to become verifiable, albeit at a relatively low absolute rate and with significant detriments in nominal test accuracy.

Summary of Contributions:

1. We propose a new type of adversarial attack to probe the undersensitivity of neural RC models to meaningful question alterations, and demonstrate the vulnerability of current models.

2. We compare two defence strategies – adversarial training and data augmentation with structurally similar unanswerable samples – and show their effectiveness at reducing undersensitivity errors on held-out data, without sacrificing standard performance.

3. We demonstrate that the resulting models generalise better in a biased data scenario with train / evaluation set mismatch.

4. We design an undersensitivity specification and apply Interval Bound Propagation to efficiently verify the specification in a Decomposable Attention Model.

5. We empirically compare the efficacy of different training and evaluation methods for formally verifying the undersensitivity specification.

1.2 Previously Published Material and Collaboration

This thesis includes experiments and findings based on papers previously published or currently under review, listed below. Individual contributions not made by the thesis author are marked at the beginning of the respective chapters.

1. J. Welbl, N. F. Liu, and M. Gardner. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 94–106, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4413. URL `https://www.aclweb.org/anthology/W17-4413`

2. J. Welbl, P. Stenetorp, and S. Riedel. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302, 2018. doi: 10.1162/tacl_a_00021. URL `https://www.aclweb.org/anthology/Q18-1021`

3. J. Welbl, P. Minervini, M. Bartolo, P. Stenetorp, and S. Riedel. Undersensitivity in neural reading comprehension, 2020b

4. J. Welbl, P.-S. Huang, R. Stanforth, S. Gowal, K. D. Dvijotham, M. Szummer, and P. Kohli. Towards verified robustness under text deletion interventions. In *International Conference on Learning Representations*, 2020a. URL `https://openreview.net/forum?id=SyxhVkrYvr`

We will continue with a general background section (Chapter 2) which discusses important aspects of RC, relevant datasets, and commonly used modelling principles. Subsequently the previously laid out research problems will be addressed in Parts I-III, after which we conclude the thesis with a summary of general findings, discussion of overall limitations, and suggestions for future work.

Chapter 2

Background

We will in this chapter discuss general background on reading comprehension, the neural modelling approach in this context, as well as commonly used dataset resources.

2.1 Reading Comprehension: Definition, Structuring, and Context

We begin with general perspectives from the field of psycholinguistics on the reading process in humans. This helps us structure and understand the high-level phenomenon of reading in some more detail, and we will tie particular aspects of it to various NLP sub-tasks, before returning again to the more high-level QA-style reading comprehension task, as found e.g. in SQUAD (Rajpurkar et al., 2016).

Reading Comprehension is an integral part of human education (Bloom et al., 1956; Krathwohl, 2002), both as a learning objective in itself, and as a means for subsequent knowledge acquisition. Literacy – the ability to read and write – has been the focus of research in psycholinguistics, and we will draw on a review from this field by Perfetti et al. (2001) to structure and illustrate various factors involved in the reading process. Following Perfetti et al. (2001) we adopt a working definition of *reading* as:

> *"the conversion of written forms into linguistic messages"* (Perfetti et al., 2001)

where a linguistic message is a modality-agnostic representation of a natural language expression – in contrast to, for example, a sign. This definition conceives reading as a process of translation: concretely, as a mapping from the space of written text into a space of linguistic messages. With this basic conception as a conversion process, reading lends itself to modeling with an input-output-based computer system (e.g. a supervised learning model) and this translation structure is reflected in a variety of reading-related NLP tasks that map text onto structured interpretations, such as semantic role labelling (Palmer et al., 2010) or compositional semantic parsing (Zelle and Mooney, 1996).

In human readers, the process of reading requires both *word identification*, and *meaning construction* (Perfetti et al., 2001). For NLP purposes however, where standardised character representations and tokenisation procedures enable the programmatic identification of words,[1] meaning construction is the main challenge.

Further following Perfetti et al. (2001), the meaning construction component itself involves three (interacting) aspects: i) the selection of contextually appropriate word meanings ii) sentence parsing iii) the integration of the message into a broader context and situation model. We will briefly expand on each of these and highlight ties to NLP.

2.1.1 Resolving Polysemy

The first of the three factors – the selection of contextually appropriate word meanings – is directly related to the NLP task of word sense disambiguation (Agirre and Edmonds, 2007). Context, redundancy, as well as implied properties can guide a reader or a computer system towards the intended sense of a word, and thus distinguish, for example, "*mouse*" as either a mammal or an electronic device.

2.1.2 Parsing

Second, parsing requires inferring the logical structure of a given textual statement, i.e. a formal representation of the relationships between its sub-components. For example, in the statement *"The mouse eats mozzarella."*, a noun (*"mouse"*, *"mozzarella"*) can be interpreted as either the subject or object of the transitive verb

[1] In this thesis we only consider English; in other languages this task is not necessarily trivial.

"eats"; part of the meaning construction process is to interpret such grammatical roles correctly.

Grammars differ in their types of composition structure – broadly separable into constituency-based, as well as dependency-based syntax – but generally agree on the hierarchical character of language and use of nested reference. Furthermore there exists a variety of semantic representation formalisms which make use of different types of basic representational structure to capture the content of a message, including predicate-argument structure, events, compositional hierarchy, as well as logical connectors (Davidson, 1967; Parsons, 1990; Steedman, 1996).

Several NLP tasks aim at extracting the semantic structure (or particular parts thereof) from text, e.g. semantic role labelling (Palmer et al., 2010), deep semantic parsing (Zelle and Mooney, 1996), information extraction (Cardie, 1997), or more focused tasks such as relation extraction (Bunescu and Mooney, 2007; Mintz et al., 2009; Riedel et al., 2010) and named entity recognition (Nadeau and Sekine, 2007). The resulting meaning representations remove both ambiguity and redundancy and have advantages when interfacing with a computer system, e.g. a query language (Zelle and Mooney, 1996; Zettlemoyer and Collins, 2005; Berant et al., 2013). Within the framework of Perfetti et al. (2001), successfully parsing a sentence leads to the *text base* interpretation: a linguistic message corresponding to the propositions explicitly stated in the text, together with a minimum of inferences required for coherent interpretation, e.g. to resolve referential coherence.

2.1.3 Context Model and Inference

The third component consists of the integration of the parsed message into a broader situational context model and inferences therein, and in this regard it differs from the text base interpretation. Reading goes beyond parsing the correct syntactic and semantic structure of the message: competent human readers are able to infer information which is not explicitly stated, and this is necessary for a richer and more flexible understanding of a document or discourse. Readers maintain a situation model which depends on previously encountered context (e.g. *"The mozzarella was poisoned."*), and this enriches the text base interpretation. Zwaan et al. (1995) suggest the five dimensions of time, space, protagonist, causality and intentionality which human readers use to form a situation model, and which they continuously update while reading.

Modelling context and inferring additional information are related to a number of different NLP tasks, each focused on a particular aspect: discourse parsing (Carlson et al., 2001) considers the broader rhetorical structure of a document; natural language inference (Dagan et al., 2006; Bowman et al., 2015) concentrates on identifying logically entailed consequences in natural language; and knowledge base inference considers inference from facts represented in predicate-argument structure – e.g. via statistical relational learning (Getoor and Taskar, 2007).

In summary, we observe that individual aspects of the decomposition of the *meaning construction* aspect of reading into three broad subcategories is mirrored in a variety of NLP tasks – each with its own history, theoretical approach, models, and evaluation benchmarks. It is worth emphasising that the above components do not form clearly bounded parts or stages, but depend on and influence each other, both in humans and in computer systems. For example, syntactic information is a useful predictive feature for word sense disambiguation (Agirre and Martinez, 2001), but can also be used to infer entailed relations between entity pairs (Riedel et al., 2013).

2.2 The Neural Approach

The decomposition laid out above highlights various aspects involved in the reading process for humans and puts into perspective the challenge of mimicking this process with a computer. The integration of these sub-tasks into a broader, high-level NLP system can be achieved in several ways: for example through the use of appropriate features, or via intermediate prediction steps in a pipeline system. Systems with concrete steps and structured representations allow for a degree of error attribution, interpretability, theoretical justification and incorporation of domain expert knowledge. Yet imposing such structure comes with several drawbacks when solving high-level NLP tasks – such as answering a comprehension question about a given text.

First, training several intermediate stages of a system pipeline independently can lead to mismatching distributions between initial component outputs, and the data which later components are trained on. Second, a pipeline may require adequate training data for each step, multiplying the annotation requirement – this may furthermore involve expert annotators, as e.g. for syntactic structure.[2] Third, erroneous predictions of initial pipeline steps may lead to problems further downstream. Later pipeline stages may fail if given faulty input, thus potentially resulting in error cascades. Fourth, the predefined choice and structure of intermediate representations may not be optimal for a given high-level NLP task at hand. Instead, using a weaker inductive model bias and very general form of intermediate representation, which is optimised based on a sufficient amount of training data, may lead to overall better model predictions.

The currently dominant paradigm of building RC computer systems overcomes these above listed issues of pipeline systems: contemporary RC approaches use *end-to-end* optimised neural network models, a highly expressive model category (Cybenko, 1989). Neural networks do not suffer from intermediate distribution mismatch, do not require annotations for intermediate stages, have no cascading errors, and can use representations optimised specifically for a given task (or – as a proxy – a particular dataset). Although reading in humans involves the previously laid out list of factors for meaning construction, there is no need in a strict sense for a computer system to mimic them explicitly, or to model them using specifically structured intermediate representations that reflect these aspects of meaning construction – if it can achieve the same functional outcome.

When used in the RC task, neural approaches generally do not structure the reading process into interpretable intermediate stages, but instead model it as a differentiable sequence of dense latent vector space transformations, which is fitted on high-level input-output pairs using gradient-based optimisation. By adapting the parametrisation of these mappings during training, neural networks can learn to as-

[2] One can alternatively rely on annotated data from a related problem, but at the cost of potential domain mismatch.

sociate input and output space in such a way, that the overall network behaviour becomes functionally equivalent to the reading process – as measured on held-out test samples of desirable input-output behaviour. As neural networks process information in a distributed fashion, their interior representations are however opaque and do not lend themselves to intepretation easily.

Large-scale datasets are a critical ingredient for training neural networks. With the availibility of these, neural networks can be trained to achieve remarkable RC abilities and have in recent years begun to outperform human performance scores on established RC benchmarks (Yu et al., 2018; Devlin et al., 2019). Note that with the paradigm shift towards general vector-based and otherwise unstructured latent variables, NLP models have lost some of their previous inductive biases. They do not use concrete pre-specified interpretation schemata and semantic relationships in their latent representations, but use vectors purely optimised to solve their given task. A system's ability to solve a given comprehension task is thus strongly influenced by the input-output data points it is trained to fit.[3] This applies consequently also to model limitations and artefacts that we observe: they can be viewed partly as a result of the particular dataset which a neural model is trained on.

With the removal of input or intermediate structure in the neural approach, we have traded away interpretability and control of the learning result for empirical performance on held out test sets. Programming desirable input-output behaviour has shifted from one extreme point of programming specific high-precision rules that prescribe desirable behaviour, to a complete relaxation of these rules into learned neural representations, where the system's programming is achieved through data.

Thus, as datasets shape the behaviour of neural RC models, both the training data and the methods to create it deserve particular research interest. In this thesis we examine the role of training data for neural machine reading comprehension systems in various scenarios, and analyse the abilities and limitations of the resulting models. With our general focus on datasets, we will to a large extent rely on previously developed neural RC models, which we will investigate in the context of

[3] Albeit arguably to a reduced extent with the more recent generation of models initialised with pre-trained representations (Peters et al., 2018; Devlin et al., 2019).

various training and evaluation datasets.

We continue with a brief overview of general neural modelling principles in the context of the language modality, e.g. pre-training, as well as different inductive biases of neural models.

2.3 Prior Knowledge and Pre-Training

2.3.1 The Role of Prior Knowledge in Reading Comprehension

An important consideration in the context of neural RC models is the implicit use of prior knowledge through pre-trained representations. Text comprehension in humans strongly depends on the existing background knowledge of a reader. Beginning already on the level of a reader's lexicon, the interpretation of a message becomes guesswork without a confident understanding of the concepts referred to. For example, without an understanding of the rules and elements of an unfamiliar sport, the comprehension of a match report mentioning e.g. an "offside trap" in football, or a "home run" in baseball, remains limited. Note that this is not a limitation to *text* comprehension in particular, but a limitation to the comprehension of concepts related to these sports in general.

For the context of Reading Comprehension, this shifts our perspective away from the message inherent in the text, and instead towards the pre-existing knowledge of the reader. Anderson et al. (1977) emphasise the role of background knowledge for comprehension in this exemplary quote:

> *"Every act of comprehension involves one's knowledge of the world as well."*

In the cognitive sciences, the role of prior knowledge to structure both comprehension and memorisation has been emphasised by schema theory (Kant, 1781; Piaget, 1926; Bartlett, 1932; Rumelhart and Ortony, 1977). Schema-theoretic views on language comprehension thus similarly stress the importance of knowledge in the reader; Adams and Collins (1977) for example state:

> *"A fundamental assumption of schema-theoretic approaches to language comprehension is that spoken or written text does not in itself*

> *carry meaning."*

But if the knowledge of the reader is such an important component to (text) comprehension, as postulated by schema theory, how can relevant structures of interpretation be made available to a computer system for RC purposes?

Several NLP-related research directions tackle the integration of background knowledge by enabling access to information from another resource. This is reflected, for example, in the research directions of information retrieval, entity linking, as well as web-scale knowledgebase construction (Banko et al., 2007; Carlson et al., 2010). With its focus on background knowledge, schema theory has however also found its way directly into AI in the form of frames (Minsky, 1974; Schank and Abelson, 1975; Charniak, 1975) and more specifically into NLP via frame semantics (Fillmore, 1982), which has inspired organised efforts to assemble large frame collections to guide schema-based interpretation in computer systems, including e.g. FrameNet (Baker et al., 1998), or the Automatic Content Extraction (ACE) project[4] which formalise schematic structures of interpretation, e.g. for particular types of events.

Formalising schemata and frames is however challenging, even definitions for what schemata and frames consist in concretely are contentious (Iran-Nejad and Winsler, 2000). Defining a comprehensive body of schemata is currently elusive, and event extraction efforts are confined to a narrow subset of possible event types and domains.

Given the difficulty of definition and limited coverage, *OpenIE* (Banko et al., 2007) uses a less rigid conception of what relations or schemata consist in, and extends relational schemata to concrete surface representations, which can be harvested automatically on a large scale. The work on universal schema (Riedel et al., 2013) broadens the notion of relational schemata further, and has empirically demonstrated that relational schemata, both in the form of concrete surface forms as well as abstract relation types, share a joint latent structure.

[4] `https://www.ldc.upenn.edu/collaborations/past-projects/ace`

2.3.2 Pre-trained Word Representations

One end point of generality and abstraction in defining schemata is to disregard particular types of relational schemata entirely, and instead only consider a very basic notion of binary relationship: the relation of two words which co-appear (or do not co-appear) in the same textual context. Naturally, such relationships are cruder than more fine-grained schematic relations which make particular assumptions on the syntactic and semantic types of their arguments, e.g. the `author_of`, or even the `is_a` relationship.

Capturing co-occurrence information for word pairs explicitly is however impractical due to the large number of possible combinations and consequent size of a matrix to store such information. In addition, low-frequency words, of which there are many (Zipf, 1935), suffer from infrequent coverage and consequently high variance in their empirical count estimates. Thus, rather than recording co-occurrence information explicitly for all word pairs in a given vocabulary, word embedding methods like *word2vec* and *glove* (Mikolov et al., 2013a; Pennington et al., 2014) approximate this information with dense vector representations (Levy and Goldberg, 2014), i.e. *word vectors*. Word vectors are optimised to predict whether word pairs co-appear in the same context, thus storing in compressed form the co-occurrence information from a large set of contexts in a given corpus. Contextual information is thus lifted into the word representations, which become to varying degrees linearly aligned with other words that they (or their typical context) are predictive of. Models using these word vectors as features then have implicit access to compressed contextual background knowledge from the corpus the word vectors were trained on. Besides lifting in background knowledge, another key advantage of dense word vector representations is their natural inter-operability with neural network models, which can be optimised towards a variety of NLP tasks. Theoretically inspired by distributional semantics (Harris, 1954; Firth, 1957), dense word vectors have thus found widespread adoption and empirical success as features in many NLP models.

Beyond computing inner products of word vector pairs, the principle of learning representations that model textual context can be realised also in more sophisticated neural networks. Indeed, pre-training neural representations on a large corpus using a language modeling objective leads to representations which prove even more empirically effective than the previous generation of word vectors (Peters et al., 2018). Especially when coupled with the transformer model (Vaswani et al., 2017), a parallelisable – and thus more efficient – neural architecture, word representations can be trained on very large text corpora to predict their surrounding context, as seen in the *BERT* and *RoBERTa* models (Devlin et al., 2019; Liu et al., 2019).

When using the resulting representations for other NLP tasks, models then implicitly have access to distributional information which is both context-dependent and estimated more precisely due to the use of larger pre-training corpora. The context-modelling information learned from a large and cross-domain text corpus is collectively stored in the model's parameters in compressed form, enabling grounded common-sense inference (Zellers et al., 2018; Devlin et al., 2019), and even allows for the decoding of concrete factual knowledge (Petroni et al., 2019; Jiang et al., 2019; Talmor et al., 2019). This compressed background knowledge learned from context modelling is implicitly available in the model's weights to support the text comprehension process, and – overall – pre-trained representations have led to the empirically strongest generation of computer systems for reading comprehension and other NLP tasks thus far.

2.4 Evaluating Text Comprehension with Questions

Thus far we have discussed the process of reading, yet not the precise task setup that a computer system is given to solve, and we shall now introduce it.

When evaluating the output of an RC computer system, it has to be compared with a correct or desirable output. Different tasks related to Reading Comprehension measure the system output's adherence to particular semantic target structures, e.g. the correct prediction of slots in Semantic Role Labeling, types of relation in Relation Extraction, or target fact triplets in Information Extraction. The evaluation of human reading comprehension ability in educational settings is however very different and sidesteps formal semantics entirely: a question (in natural language) is posed to the student that specifies a particular information request about the given text, which is answered again in natural language.

The Reading Comprehension task in NLP follows a similar paradigm. Concretely, a given document d (usually a document paragraph) is provided to a model, together with a comprehension question q. The model then predicts an answer to the question q which is compared to the correct answer a, that usually has been established through manual data annotation efforts. We emphasise again that generally in this task d, q, and a are natural language expressions, although variations and deviations exist.

Question-based comprehension evaluation has found widespread adoption and several datasets created in recent years make use of this design choice (Richardson et al., 2013; Rajpurkar et al., 2016). The approach is very flexible: arbitrary aspects of the information in the text can be queried in q, which is not bound to a particular schema of interpretation or composition structure. At the same time, understanding the information request formulated in the question q requires itself the comprehension of a natural language expression.

Given the limitations in generating free-form text as well as the inherent challenges associated with its evaluation, RC systems are in many cases offered a restricted set of possible answers: they are either given questions in *multiple-choice* format, or an *extractive* task format, in which the answer to the comprehension question consists of a usually short text span within the given document.

In contrast to annotating language with rich linguistic structure (Marcus et al., 1993), the paradigm of posing natural language comprehension questions has the advantage of lending itself to non-expert data annotation. This allows for cost-efficient annotation crowdsourcing (Snow et al., 2008; Rajpurkar et al., 2016) at a large scale, thus enabling the creation of datasets of a size suitable to fit parameter-rich neural RC models.

Data collection and neural modeling are both essential components to the current paradigm of building RC systems; we will briefly discuss both in the following two sections.

2.5 Neural Models for Reading Comprehension

Following the advent of pre-trained word representations (Mikolov et al., 2013a; Pennington et al., 2014) and large-scale datasets (Hermann et al., 2015; Rajpurkar et al., 2016), a profusion of neural model architectures has been developed for the RC task (Weissenborn et al., 2017; Seo et al., 2017a; Yu et al., 2018). These models make use of dense latent representations, and the set of functions to compute them is parameterised in a high-dimensional vector space. Parameters are optimised end-to-end in a way that associates desirable input-output pairs, thus forming connections between each pair of comprehension question q and document d, with the corresponding answer a.

Neural models in NLP generally differ in their architectural design, and the choice of the particular type of function to connect inputs with outputs determines the inductive bias of the resulting model. Convolutional layers, for example, can model translation-invariant local textual patterns in a document; they represent an efficient model category that has proven useful e.g. in text classification (Kalchbrenner et al., 2014; Kim, 2014) or to address unknown words at the character level (Seo et al., 2017a). Sequential encoders on the other hand, such as RNNs, LSTMs (Hochreiter and Schmidhuber, 1997) or GRUs (Cho et al., 2014), encode text following the temporal order of language. Since the effective depth of their computation depends on the length of the sequence, they are less efficient, and have difficulty in capturing long-range dependencies. Finally, attention-based architectures facilitate conditioning across long distances in a text, and can be parallelised across the temporal axis of the text sequence. They can be applied either on the output of sequential encoders (Hermann et al., 2015; Rocktäschel et al., 2016), or as the central architectural components itself, as in the transformer architecture (Vaswani et al., 2017).

The above neural architecture types are frequently combined with one another, and are generally used across NLP tasks and not only for RC. In particular for RC however, a general and widely used modelling recipe is to i) encode tokens using pre-trained word representations, e.g. *GloVe*; ii) process the resulting representations with one of the aforementioned neural layer types to allow for interaction; iii) to predict output probabilities for different answer options, e.g. using a softmax probability distribution derived from the representations of different start and end tokens of possible answer spans in the given document.

This general recipe is followed, for example in the BiDAF (Seo et al., 2017a), FastQA (Weissenborn et al., 2017) or QANet (Yu et al., 2018) models, each with slightly different architectural details. Note that BERT (Devlin et al., 2019) combines i) and ii) and extends the pre-training of representations into the deeper layers of the network.

Throughout this thesis we will make use of several neural RC models for our experiments. We direct the reader to the original published works on these models – which will be referenced in the relevant sections – for concrete details and underlying design choices of particular architectures.

2.6 Reading Comprehension Datasets

A variety of dataset resources exists to both train and evaluate reading comprehension models. These datasets differ in their precise task formulation, complexity, number of datapoints, domain, and annotation methodology.

One early high-level Reading Comprehension dataset is MCTEST (Richardson et al., 2013). This dataset contains short fictional stories and comprehension questions with *multiple choice* answers, which were collected using crowdsourced data annotation. It demonstrated a scalable approach for gathering comprehension questions, yet with 2,000 questions the dataset was insufficiently large to serve as a resource for data-driven neural network architectures.

Other datasets were proposed which leverage document structure to train and measure text comprehension. For example, the CHILDRENBOOKTEST dataset (Hill et al., 2016) formulates questions in *cloze* style, or the CNN/DAILYMAIL dataset (Hermann et al., 2015) leverages the observation that news texts contain brief textual summaries about their content. Exploiting document structure to automatically gather a large number of data points from a corpus comes at no annotation cost, and the resulting datasets have proven valuable in the development of early RNN-based RC models. On the other hand, the resulting samples can be noisy, deviate from the above described RC task structure, and models quickly reached the bounds on their possible performance (Chen et al., 2016).

Crowdsourced data annotation was successfully applied at a larger scale for the related task of Natural Language Inference in the SNLI dataset (Bowman et al., 2015), demonstrating the scalability of the approach with more than 500,000 annotated samples. This, together with the general penetration of the NLP field by data-hungry neural networks ultimately led to the seminal work on the SQUAD dataset (Rajpurkar et al., 2016). SQUAD1.1 contains high-quality natural language comprehension questions about Wikipedia paragraphs chosen across a variety of topics. The availibility of this large dataset with more than 100,000 samples, together with maturing neural modelling toolkits, such as *theano* (Theano Development Team, 2016), *tensorflow* (Abadi et al., 2015), and *pytorch* (Paszke et al., 2019), led to a profusion of neural network models; the SQUAD leaderboard[5] bears witness to this remarkable collective research effort. While crowdsourcing had been used before (Snow et al., 2008; Richardson et al., 2013; Bowman et al., 2015), SQUAD was the first realisation of this approach for RC on a large scale. Created by lay annotators, the SQUAD dataset has enabled the training and benchmarking of neural RC models which have grown progressively more capable, and which today exceed human performance metrics.

Mixed into the general enthusiasm about the apparent progress and empirical success of the neural and data-driven approach to building RC models, voices of caution were however soon raised. For example, Jia and Liang (2017) demonstrated that although achieving strong test set generalisation, models trained on SQUAD1.1 fail dramatically when appending adversarially chosen sentences to

[5] `https://rajpurkar.github.io/SQuAD-explorer/`

the given paragraph. Clearly RC had not been solved yet, despite the 'super-human' performance measures on the hidden held-out test set.

One lesson learned from the Jia and Liang (2017) paper was that it is necessary to include *unanswerable* comprehension questions into a dataset. Otherwise, models can quickly learn that type-consistency clues (Sugawara et al., 2018) – e.g. a date for a *When* question – can be sufficient to pick the correct answer span in the document. In its second iteration, the SQUAD2.0 dataset (Rajpurkar et al., 2018) thus includes more than 40,000 crowdsourced unanswerable samples into the existing dataset, and encourages researchers to develop methods which can predict whether a comprehension question is answerable or not, given the text. NEWSQA (Trischler et al., 2017) is a second dataset which follows this principle: covering the news domain of the previous CNN/DailyMail datasets, NEWSQA samples contain natural language comprehension questions and notably also include unanswerable questions.

After demonstrating the effectiveness of collecting large-scale RC datasets to train neural RC models, dataset creation was extended to new domains, data sources, and comprehension phenomena. Reading Comprehension datasets were created, for example based on reading comprehension exam questions (Lai et al., 2017), book and movie plots (Kočiský et al., 2018; Saha et al., 2018), cooking recipes (Yagcioglu et al., 2018), comprehension questions involving discrete operations such as comparisons (Dua et al., 2019), as well as in connection with a QA system for web engine search queries (Nguyen et al., 2016; Kwiatkowski et al., 2019). Increasingly, research also addresses cross-dataset generalisation, in order to avoid overfitting to particular domains and dataset setups (Talmor and Berant, 2019).

In this thesis we will add to this body of research and describe our own experience with the development of new RC datasets. The first dataset (Part I) is inspired by the success of SQUAD1.1, yet aims at a new domain: it was constructed by crowdsourcing the creation of questions in the science exam QA domain. This comes with a unique set of challenges, but holds the promise of supporting a computer system that solves real science exams with training data to train relevant RC capabilities. The second dataset (Part II) aims at overcoming a limitation of the SQuAD dataset: that it revolves relatively close to the *text base* interpretation. That is, by posing questions with a considerable degree of lexical overlap, SQuAD questions address the explicit interpretation of the given paragraph, and answer-relevant information can mostly be found within a single sentence (Min et al., 2018). Instead we will work on broadening the RC task towards a multi-hop setting, in which textual information from several sources has to be combined to infer the answer to a comprehension query. This reduces the degree of lexical overlap and involves interpretation beyond the text base, yet dataset creation for such a task is non-trivial. Finally, in Part III we will investigate a data-induced limitation to neural RC models. Commonly used RC datasets, concretely SQuAD2.0 and NewsQA, lack unanswerable questions that are structurally similar to existing answerable questions, which results in models that lack sensitivity to critical aspects of the information requested in the comprehension question.

Throughout these following three parts of the thesis, our aim is to advance progress in RC with experimental insights and the provision of data resources to the research community. We will highlight particular aspects of the dataset creation procedure and demonstrate at various points how it can result in measurable artefacts visible in models trained on the resulting datasets. Our hope is that future efforts in RC dataset creation can take inspiration both from the set of methods used, dataset biases discovered, as well as experimental findings on the resulting datasets.

Part I

Machine Reading Comprehension for Science Exams

Chapter 3

Dataset Assembly for Reading Comprehension in Science Exam Questions

> *The content of this chapter is based on previously published work (Welbl et al., 2017). The chapter includes results of experiments conducted not by the author of this thesis, but by collaborators at the Allen Institute for Artificial Intelligence, which has developed a variety of model implementations for science exam QA. Concretely, this refers to the first five rows in Table 3.2, to Table 3.3, and to the results on* BiDAF *(direct answer setting).*

Answering natural science school exam questions is not only a challenge to humans, but also to NLP systems. It involves question comprehension, the identification and extraction of relevant textual or structured information to support possible answers, common sense reasoning, as well as the integration of these and several other abilities (Clark et al., 2013; Clark, 2015). For example, in order to solve the question *"With which force does the moon affect tidal movements of the oceans?"* a model must both interpret the information request formulated in this question, possess an understanding of an abstract natural phenomenon, and be able to link the information request with this background knowledge.

Prior to this study, in a 2016 competition for developing systems that solve 8th

Q: When a meteoroid reaches earth, what is the remaining object called?

A: **meteorite**
B: comet
C: meteor
D: orbit

Textbook Passage: Meteoroids are smaller than asteroids, ranging from the size of boulders to the size of sand grains. When meteoroids enter Earth's atmosphere, they vaporize, creating a trail of glowing gas called a meteor. If any of the meteoroid reaches Earth, the remaining object is called a meteorite.

Figure 3.1: Natural phenomena are frequently taught in educational curricula and described in many standard school text books. We use such passages as the basis for multiple choice question generation in the science exam domain, resulting in samples like the one in this figure. In total, 13,679 such questions were collected.

grade multiple choice science exam questions (Schoenick et al., 2016), the strongest method achieved a score of 59.3% accuracy; Information Retrieval (IR) was shown to be a very strong baseline (Clark et al., 2016); yet neural RC systems were not among the top-scoring methods.[1] Whereas QA and RC system performance in other domains has substantially advanced with the use of high-capacity neural network models (Kadlec et al., 2016; Dhingra et al., 2016; Sordoni et al., 2016; Seo et al., 2017a), one factor that has impeded progress of neural RC methods in science exam QA is the lack of large, in-domain training resources. This raises the question how we can collect such a dataset for the science exam domain, and thus potentially support the RC capabilities of neural science exam solution approaches.

The aim of this chapter is to develop a training and evaluation resource for the science exam QA domain. We will describe a data annotation method which involves both the formulation of new questions and answers, as well as the creation of plausible false answer candidates – supported by model suggestions. We crowdsource the creation of these multiple-choice questions using text passages from educational materials and gather a total of 13,679 multiple choice questions with a budget of $10,415. The resulting dataset will be referred to as SciQ; Figure 3.1 shows one of its example questions.

This resulting dataset can be used for a multiple choice QA task, in which the

[1] Note that in more recent work, which was published after this study was conducted, the picture has changed and models with scores up to 90.7% have been developed (Clark et al., 2019), notably with the introduction of neural RC methods, and in particular large-scale pre-trained language models (Peters et al., 2018; Devlin et al., 2019).

goal is to predict an answer among several options using any relevant background knowledge a system can potentially identify, e.g. using information retrieval. Alternatively, it can be used for a direct-answer task, in which a model is given the question and has to predict an answer as a span in the given text passage. These two dataset versions allow for research both on the integration of RC systems with retrieval components in the science exam domain, as well as more focused studies on RC in isolation, when relevant text is already provided.

After a detailed description of the dataset assembly process, we will discuss several experiments with previously developed neural RC models on SciQ, as well as other established science exam solvers. We will both compare the performance of different methods on the newly assembled dataset and discuss its usefulness as additional training resource for solving real exam questions. The central research questions in this chapter will thus be the following:

List of Research Questions Addressed in this Chapter:

1. How can the crowdsourcing approach to RC dataset creation be adapted for collecting a dataset in the science exam domain?

2. How do previously developed science exam solvers and neural RC methods perform on the resulting dataset?

3. Can the resulting data be used as additional training resource to improve the performance of RC systems on science exam questions?

3.1 Annotation Method Overview

Constructing a sizable dataset of multiple-choice science questions, ideally similar to real exam questions, poses a set of unique challenges. Like in SQUAD (Rajpurkar et al., 2016), annotation workload can be distributed by crowdsourcing the task, yet annotators generally cannot be expected to possess the same degree of domain knowledge as teachers or the designers of educational curricula. Furthermore, as we consider a multiple-choice QA setting, questions with poorly chosen false answer candidates can be trivial to solve.

Our data collection pipeline comprises the following main steps: we first use a noisy filter to select potentially relevant text passages from study books, of which we then show multiple options to annotators to choose from when composing questions. Next, we select high-confidence predictions for false answer candidates, given by a model trained on a set of real multiple-choice exam questions to predict plausible false answers. These predictions are then used to support annotators in converting the previously produced questions into multiple choice QA examples.

Overall the procedure thus exploits both an existing corpus of in-domain study materials and a smaller set of existing exam questions. The method broadly follows the crowdsourced annotation setup of SQUAD, in which the annotation task requires reading a text passage and formulating a question about it. However our approach focuses on passages from the domain of science textbooks in particular, as questions may otherwise lack both relevance and topic variety, and we further deviate from the SQUAD setting by constructing a multiple-choice QA dataset with several plausible answer distractors. By supporting annotators with sets of passages and model predictions, the human intelligence task is modified from an exclusively *generative* task of producing questions from scratch – which is difficult, slow, expensive, and can result in a lack of diversity when repeated – to a task that requires the user to *select*, *modify* and *validate* – which is less challenging, faster, more cost-effective, and with variation in content induced by the given paragraphs and model suggestions.

3.2 Relation to Prior Work

Dataset Construction A series of QA datasets has been created prior to this work, e.g. based on Freebase (Berant et al., 2013; Bordes et al., 2015), Wikipedia (Yang et al., 2015; Rajpurkar et al., 2016; Hewlett et al., 2016), web search queries (Nguyen et al., 2016), news (Hermann et al., 2015; Onishi et al., 2016) and books (Hill et al., 2016; Paperno et al., 2016), and we add to this work by constructing a dataset for the science exam domain. In contrast to some of the prior datasets, SciQ contains natural language questions composed by annotators, instead of cloze questions (Hermann et al., 2015; Hill et al., 2016). In addition, it focuses on the creation of multiple-choice questions, which includes the selection of plausible answer distractors.

Science Exam QA Clark et al. (2013) give a broad overview of the particular challenges in the multiple-choice science exam QA task, as well as possible solution approaches. Generally, prior work on this task varies in their methodology, and often focuses on particular sub-problems. For example, Li and Clark (2015) enrich questions with additional structured background information and evaluate the coherence of the resulting scenes, whereas Sachan et al. (2016) model entailment using derivations from knowledge items matched with max-margin ranking. Other work uses Markov logic networks (Khot et al., 2015), or integer linear program (ILP)-based methods to construct chains that derive the answer from structured background knowledge (Khashabi et al., 2016). In contrast to these methods which derive answers symbolically via several steps, the dataset we will assemble aims less at complex inference skills, but more at the text comprehension skills necessary to interpret questions and potentially relevant passages. The *Aristo* ensemble (Clark et al., 2016) mixes several other solution approaches with complementary strengths, including both symbolic reasoning approaches, retrieval, and a shallow statistical approach based on word co-occurrence. Finally, neural approaches to science exam QA have – prior to the assembly of this dataset – not found much adoption, likely due to the lack of adequate data resources. With the effort described in this chapter we address this problem: we assemble a dataset larger than previously available collections, and we will discuss baseline results of both established science exam solvers and neural RC methods previously used for other datasets.

Automatic Question Generation Prior work has addressed the automatic conversion of declarative statements into questions, mostly in didactic contexts. While some methods use syntactic templates for this transformation (Mitkov and Ha, 2003; Heilman and Smith, 2010), another approach is to leave gaps in the text to form cloze questions. Initially we considered these approaches, but we observed that for our purpose these automatic methods did not suffice in producing high-quality samples. In our target application, we furthermore address a multiple-choice setting, in which several plausible answer distractors have to be defined. For this particular task of predicting plausible answer distractors, prior work has proposed a number of similarity metrics (Mitkov et al., 2009), which includes metrics based on WordNet (Mitkov and Ha, 2003), a thesaurus (Sumita et al., 2005) or distributional information (Pino et al., 2008; Aldabe and Maritxalar, 2010). Other work relies on domain-relevant ontologies (Papasalouros et al., 2008), morphological or phonetic similarity (Pino and Eskénazi, 2009; Correia et al., 2010), probabilities for the context of the question (Mostow and Jang, 2012) and context-dependent lexical entailment (Zesch and Melamud, 2014). Instead of similarity-based answer distractor selection, our approach relies on a trained model to suggest answer distractors, for which we will exploit several of the previously named heuristics as features. Prior work in the context of biology questions has also relied on features to predict answer distractors (Agarwal and Mannem, 2011; Sakaguchi et al., 2013). The model we choose uses a random forest for candidate ranking, can work both for questions composed by humans as well as cloze-style questions, and is targeted specifically at answer distractors for science exam questions.

3.3 Method: Assembling a Science Exam QA Dataset

In this section we lay out a method to assemble a multiple-choice science exam QA dataset, which comprises two annotation stages. First we give a crowd worker a set of short text passages from a base corpus, and ask them to pick one to formulate a question about. Second, a different annotator is shown the QA pair produced in the previous step. This annotator has the option to reject previously written questions, and then enters three false answer candidates, supported by the suggestions of an answer distractor prediction model. The outcome of this process is a multiple-choice question consisting of both a question q, a supporting text passage s, and a set C of candidate answers, where $a^* \in C$ is the correct answer. We will next focus in more detail on these two individual steps of the dataset creation process.

3.3.1 Step 1: Producing In-domain Questions

Base Corpus The choice of a relevant text corpus as the basis for the composition of questions is an important factor for the resulting dataset characteristics. In order to create science questions, the base corpus should be aligned with topics of school exams, yet not be too specific, linguistically complex, or technical (e.g. scientific papers). Documents retrieved from the web when searching for content keywords related to science exams (e.g. *"animal"* or *"food"*) contain a substantial fraction of irrelevant documents, often with commercial content. On the other hand, Simple Wikipedia articles from science-related categories contain more factual information, but often include very specific facts (e.g. *"Hoatzin can reach 25 inches in length and 1.78 pounds of weight"*). Instead, we choose study textbooks as our base corpus: their content is both relevant and directly tailored towards a student audience. While the digital availability of such study resources is limited, we identified a collection of 28 books from several resources related to online learning, such as CK-12[2] and OpenStax[3] – most of which have shared the materials with a Creative Commons License. These digitally available books cover topics in chemistry, physics, biology and earth science, and range from elementary to college introductory level. They contain descriptions of general phenomena covered in natural science education, instead of cataloguing detailed, yet overly specific knowledge – as e.g. the above mentioned fact about the Hoatzin. In Appendix A we list titles and sources for all books used in this base corpus.

[2] `www.ck12.org`

[3] `www.openstax.org`

Paragraph Filter Not all documents within this large corpus of study books are relevant or appropriate to write questions about: study books contain an abundance of unsuitable text such as instructions, references to other material, or lengthy illustrative examples. To narrow down the base corpus to a smaller set of more relevant text passages, we filter out individual paragraphs according to a set of rules which we next describe. Our filter comprises both lexical, syntactic, and pragmatical rules, as well as constraints based on sentence complexity. Concretely, the filter considers individual sentences, and filters them out if they *i)* are an exclamation or question *ii)* have no verb phrase *iii)* contain imperative phrases. This serves the goal of removing non-declarative statements. Furthermore, we filter out sentenced which *iv)* contain demonstrative pronouns *v)* begin with a pronoun *vi)* mention a graph, table or web link *vii)* begin with a discourse marker (e.g. *"Nonetheless"*) *viii)* contain modal verbs. These rules are included because they often mark statements that rely on further context, which we want to avoid for subsequent annotation stages. In addition, statements are removed that focus on aspects of teaching but are unrelated to the content; concretely statements that *ix)* contain instructional vocabulary (*"teacher"*, *"worksheet"*) *x)* contain personal pronouns other than the third-person (as often used in instructions) *xi)* contain absolute wording (e.g. *"never", "nothing", "definitely"*) *xii)* contain first names (to avoid illustrative stories with a hypothetical scope "*Karina has a bicycle...*"). Finally, we filter out sentences that *xiii)* have less than 6 or more than 18 tokens or more than 2 commas *xiv)* contain special characters other than punctuation *xv)* have more than three tokens beginning uppercase. These rules set limits to the syntactic complexity of the sentences, and remove undesirable formatting artefacts and acronyms that may require additional context. This overall filter is applied on a per-sentence level on the full corpus of study books. The system then removes paragraphs for subsequent annotation which do not at least possess a minimum of one sentence passing the filter.

Discussion Several of the above heuristics are applicable more generally to help identify simple and declarative sentences from a text corpus. Yet they cannot by themselves ensure domain relevance, e.g. for science exams – this is achieved with the choice of the base corpus (in our case: the selection of study books). Furthermore, the above heuristics are unreliable, and the set of artefacts pointed out above (e.g. instructions or references to figures) is not comprehensive. The artefacts we highlighted are a common phenomenon in the educational material we choose, but likely not as relevant in other domains. Next, the above heuristics are mostly shallow, and thus miss cases which cannot be detected, e.g. with a lexical match. For example, a reference in a passage might still point to content outside of it, even though we have a filter for demonstrative pronouns and keywords for graphs, tables, and web links. In summary, the application of these heuristics is not a guarantee that undesirable passages are filtered out; they are instead intended to reduce their total prevalence and result in fewer irrelevant passages, but also come with potential false negatives.

Question Formulation Task After having chosen a base corpus and filtered its individual passages, we next provide these passages to crowd annotators and collect questions about their content. While many irrelevant passages have been filtered out with the previously described filter heuristics, the resulting documents still contain a considerable fraction of irrelevant passages. We thus give *three* passages to each annotator and provide them with the option to choose any one of them, or to reject all if the material is deemed irrelevant. In the annotation guidelines we further specified the desirable attributes of the questions that are to be formulated: *(i)* no questions with *yes/no* answers *(ii)* questions should be about the text and not require additional information beyond that *(iii)* where possible, questions should address general principles instead of particular factual details *(iv)* the length of questions should preferably be between 6 and 30, and the answer up to 3 tokens *(v)* ambiguous questions should be avoided, and *(vi)* the correct answer should be clear given the content of the chosen paragraph. In the annotation guidelines, we showed examples of both desired and undesired queries, alongside explanations for what makes them good or bad. Crowd annotators were given the ability to contact us and were encouraged to give feedback, which several annotators made use of when we conducted the annotation. Advertising the task on *Amazon Mechanical*

Turk, we offered 0.30$ compensation per written question. In total, 175 annotators participated in the project, and 12.1% of cases were rejected as all three documents were deemed irrelevant.

Discussion The rejection rate of 12.1% is a lot smaller compared to a task setup in which only a single passage were given, assuming the same underlying passage set. Presenting multiple short passages at once increases the probability of including at least one suitable text, it is thus more economical and in the ideal case it also helps to match the preferences of individual annotators. A potential drawback of this approach is that it may result in an overall tendency and consequent data bias towards "popular" topics, although this is difficult to measure. Next, questions created using crowdsourced annotation can suffer in quality compared to annotations produced by a small set of expert annotators. To mitigate this risk we require *Master's* status among the annotators, and furthermore include a validation task for the resulting questions in the subsequent annotation stage. Finally, another feature of crowdsourced questions is that the resulting questions are relatively close to what is explicitly stated in the passage, e.g. the passage "*Without Coriolis Effect the global winds would blow north to south or south to north. But Coriolis makes them blow northeast to southwest or the reverse in the Northern Hemisphere. The winds blow northwest to southeast or the reverse in the southern hemisphere.*" results in the question *"What phenomenon makes global winds blow northeast to southwest or the reverse in the northern hemisphere and northwest to southeast or the reverse in the southern hemisphere?"* with the answer *"Coriolis Effect"*. Formulating questions this close to the text is clearly a desirable feature if the underlying text explicitly describes directly relevant natural phenomena; it is however also a limitation as the necessary text comprehension frequently resides on the text base level (cf. Section 2.1.2 and 2.1.3). The resulting questions thus often possess a considerable degree of lexical overlap with the passage, and this may lead to over-estimated capabilities of IR baselines for these questions compared to actual exam questions, if given access to a relevant text corpus.

3.3.2 Step 2: Selecting Answer Distractors

Following the collection of question-answer pairs described in the previous section, we will next lay out a method for adding alternative answer candidates as distractors. During an initial annotation trial conducted by ourselves, we observed that generating plausible false answer options poses a substantially higher time demand than formulating a question about a given passage. To facilitate this process, we thus provide model-based suggestions for answer distractors in the next step of the annotation task. This exposes annotators to relevant suggestions, and some of the suggestions can be accepted directly if deemed sufficient. We will continue first with a discussion of desirable attributes for false answer candidates, then define and train a model for proposing false candidates, and finally describe how they are leveraged in the second annotation step.

Desirable Distractor Characteristics When generating multiple-choice answer candidates, it is critical that the alternative candidates are convincing. Multiple choice questions with unrelated or nonsensical false answer candidates pose a substantially easier task than questions with plausible candidates. The former would likely be less useful as a resource to train models for science exams, as a model could solve questions by excluding nonsensical candidates. The main challenge in generating multiple choice answer candidates is then not the identification of false answer expressions to q, but the identification of expressions that are *plausible* answer candidates. Apart from being incorrect answers, we identify the following list of desiderata for alternative answer candidates:

- grammatical consistency; e.g. for the question "*When animals use energy, what is always produced?*" the expected answer is a noun phrase.
- consistency according to abstract attributes: if the correct answer a^* is a member of a particular category (e.g. a^* is a chemical element), then good alternative answer options fall into the same or a similar category.
- consistency with the question topic: a question about oceans and ecosystems should ideally have other answer distractors than e.g. *"aikido"* or *"soprano"*.

Our model to automatically predict plausible answer distractors utilises a feature representation that takes into account a variety of information, including features related to the above desiderata. With access to these features, the model is fitted on a set of actual multiple-choice science exam questions to learn properties of plausible distractors, and ideally rank them above implausible ones. We next introduce this model to generate plausible answer distractors in more detail.

Distractor Model Overview On a fundamental level, the model ranks potential answer candidates from a large set of expressions $\bar{C}$ and chooses the highest scoring elements. It uses a ranking function

$$r : (q, a^*, c) \mapsto s_c \in [0, 1] \tag{3.1}$$

which produces a score s_c used to determine if $c \in \bar{C}$ is a plausible false answer in the particular context of the query q and the correct answer a^*. To define the ranking function r we use the score $s_c = P(c \text{ is plausible} \mid q, a^*)$ stemming from a binary classification model trained to distinguish plausible candidates from other, random expressions, using a set of features $\phi(q, a^*, c)$ derived from q, a^*, and c. This classification model is trained on a small set of real multiple choice exam questions, where we use the false answers as plausible examples, and a set of randomly chosen expressions from $\bar{C}$ as negatives, which we sample in equal proportion. We opt for a random forest (Breiman et al., 1984; Breiman, 2001), a classifier with robust generalisation performance for small and medium-sized datasets, and the capacity to model nonlinear interactions. We next list the features $\phi(q, a^*, c)$ available to this classifier; they are derived from both the question q, the correct answer a^* and a tentative distractor expression $c \in \bar{C}$. Using these features, the model can learn common attributes of plausible distractors observed in original science exam questions and – based on the patterns learned – propose false candidates that appear realistic for new (q, a^*) pairs.

Distractor Model: Feature List We now list the features used by the false candidate prediction model:

1. Bags of *GloVe* (Pennington et al., 2014) embeddings for q, a^*, and c;
2. An indicator for PoS-tag consistency of a^* and c;
3. Singular / plural consistency of a^* and c;
4. Logarithm of average word frequency in a^* and c;
5. Levenshtein string edit distance between a^* and c;
6. Suffix consistency of a^* and c (e.g. for (*"regeneration", "exhaustion"*));
7. Token overlap indicators for q, a^* and c;
8. Token and character length for a^* and c, and similarity therein (based on relative proportions);
9. Indicators for numerical content in q, a^*, and c, and consistency therein;
10. Indicators for units of measure in q, a^*, and c, and for co-occurrence of the same unit;
11. WORDNET-based hypernymy indicators between tokens in q, a^*, and c, in both directions, and potentially via two steps;
12. Indicators for 2-step connections between entities in a^* and c via a KB based on *OpenIE* triples (Mausam et al., 2012) extracted from pages in Simple Wikipedia about anatomical structures;
13. Indicators for shared WORDNET-hyponymy of a^* and c to one of the concepts most frequently generalising all three question distractors of real science questions (e.g. *element*, *organ*, *organism*).

The features involving KB links and indicators for hypernymy can describe sibling structures between a^* and c based on a common attribute or hypernym. If, for example, the correct answer a^* to a question is *heart*, then one plausible alternative answer candidate might be *liver*, which shares the hyponymy relation to *organ* with a^*.

Distractor Model Training The distractor prediction model is trained on a total of 3,705 multiple-choice questions from 4th and 8th grade science exams, of which we use 80% for training and 20% for validation. Each of the samples from this dataset contains four answer candidates, i.e. three examples of false candidates. Using the `scikit-learn`'s random forests implementation with default hyperparameters, we train a model with 500 trees, enforcing a minimum of 4 examples per decision tree leaf. The same set $\bar{C}$ is used both to draw random samples during training that are contrasted against the observed distractors, and as the set of entries to rank, for new questions at test time. That is, at test time all potential candidates $c \in \bar{C}$ are ranked for a new given (q, a^*) pair, and the highest-scoring elements will become the model suggestions. The set $\bar{C}$ contains a total of 488,819 expressions, concretely: (1) the 400,000 tokens in the pre-trained *GloVe* vocabulary; (2) any answer candidate from the set of training questions; (3) a collection of noun phrases gathered from Simple Wikipedia articles about body parts; (4) $\sim$6,000 additional noun expressions collected from primary school texts in the science subject. We furthermore add expressions formed by replacing any one token in a^* with a unigram in $\bar{C}$, in the case of samples for which a^* is a multi-word expression.

Distractor Model Evaluation The random forest model for predicting false answer candidates achieves 99.4% and 94.2% accuracy on the training and validation set, respectively, where we measure the binary classification accuracy of distinguishing real answer distractors from randomly drawn alternatives from $\bar{C}$. We show examples for the highest scoring predictions of this distractor model in Table 3.1. In a qualitative inspection, we observe that many of the high-ranking model suggestions are indeed plausible answer distractors. When considering failure cases, the predicted distractor is often semantically related, but from a different abstract category (for example *"nutrient"* and *"soil"* in column 1 of Table 3.1 are not chemical elements). For other examples the level of specificity is misaligned (e.g. in column 3: *"frogs"*). We also observe that multi-word expressions are more likely to be ungrammatical or irrelevant, despite the inclusion of PoS features; we note that these can themselves occasionally be erroneous and potentially result in cascading errors.

Q: Compounds containing an atom of what element, bonded in a hydrocarbon framework, are classified as amines?	**Q:** Elements have orbitals that are filled with what?	**Q:** Many species use their body shape and coloration to avoid being detected by what?	**Q:** The small amount of energy input necessary for all chemical reactions to occur is called what?
A: nitrogen	**A:** electrons	**A:** predators	**A:** activation energy
oxygen (0.982)	**ions** (0.975)	**viruses** (0.912)	conversely energy (0.987)
hydrogen (0.962)	atoms (0.959)	ecosystems (0.896)	**decomposition energy** (0.984)
nutrient (0.942)	crystals (0.952)	frogs (0.896)	membrane energy (0.982)
calcium (0.938)	protons (0.951)	distances (0.8952)	motion energy (0.982)
silicon (0.938)	neutrons (0.946)	**males** (0.877)	context energy (0.981)
soil (0.9365)	**photons** (0.912)	crocodiles (0.869)	**distinct energy** (0.980)

Table 3.1: Selected distractor prediction model outputs. For each QA pair, the top six predictions are listed in row 3 (ranking score in parentheses). Boldfaced candidates were accepted by crowd workers.

Question Validation and Distractor Selection Task We now proceed to the final annotation step, in which multiple-choice answer candidates are added to the previously written questions. We give the result of the first annotation task – a (q, a^*) pair – to a crowd annotator, alongside the six highest scoring predictions for false answer candidates formed by the above described model. This task has two goals: first, to control the quality of previously written questions; and second, to validate the false answer distractors suggested by the model, or compose new alternatives if they are insufficient. The annotation instructions were given as follows: first, to determine whether the given question would be likely to appear in a school exam of a science subject; each question could at this point be labelled as unrelated to science, ungrammatical, having an incorrect answer, or necessitating very specific background knowledge. A total of 92.8% of the previously collected questions passed this stage. A second instruction was then given to pick a maximum of two of the six suggested false answer candidates, whereas at least one false answer had to be written by themselves, resulting in a total of three false answer candidates. We added this requirement for annotators to write an answer candidate themselves after an initial pilot annotation task, in which we observed that it helped engage the annotators and resulted in a higher quality of false answer candidates. Again, we provided annotators with several examples of desirable, as well as undesirable false answer candidates, and the opportunity to give feedback on the annotation task. This annotation task was advertised on *Amazon Mechanical Turk*, paying 0.20$ per completed annotation task, and again only working with annotators in possession of AMT *Master's* status. Annotators found the suggested distractors to be sufficient in about half of the cases; 36.1% of the false answer candidates in the final dataset are model-generated.[4] We observe that acceptance rates for the model-suggested candidates are generally higher for short answers, whereas almost none of the predictions were accepted for the few questions with answers that are very long.

This concludes the description of the dataset assembly procedure. The remainder of this chapter will discuss characteristics of the resulting SciQ dataset, establish benchmark performance for several systems, as well as a human performance baseline on the questions gathered.

[4] As annotators were able to pick at most two model-suggested candidates, the maximum is 66%.

Example 1	**Example 2**	**Example 3**	**Example 4**
Q: What type of organism is commonly used in preparation of foods such as cheese and yogurt?	**Q:** What phenomenon makes global winds blow northeast to southwest or the reverse in the northern hemisphere and northwest to southeast or the reverse in the southern hemisphere?	**Q:** Changes from a less-ordered state to a more-ordered state (such as a liquid to a solid) are always what?	**Q:** What is the least dangerous radioactive decay?
1) mesophilic organisms 2) protozoa 3) gymnosperms 4) viruses	1) coriolis effect 2) muon effect 3) centrifugal effect 4) tropical effect	1) exothermic 2) unbalanced 3) reactive 4) endothermic	1) alpha decay 2) beta decay 3) gamma decay 4) zeta decay
Mesophiles grow best in moderate temperature, typically between 25°C and 40°C (77°F and 104°F). Mesophiles are often found living in or on the bodies of humans or other animals. The optimal growth temperature of many pathogenic mesophiles is 37°C (98°F), the normal human body temperature. Mesophilic organisms have important uses in food preparation, including cheese, yogurt, beer and wine.	Without Coriolis Effect the global winds would blow north to south or south to north. But Coriolis makes them blow northeast to southwest or the reverse in the Northern Hemisphere. The winds blow northwest to southeast or the reverse in the southern hemisphere.	Summary Changes of state are examples of phase changes, or phase transitions. All phase changes are accompanied by changes in the energy of a system. Changes from a more-ordered state to a less-ordered state (such as a liquid to a gas) are endothermic. Changes from a less-ordered state to a more-ordered state (such as a liquid to a solid) are always exothermic. The conversion . . .	All radioactive decay is dangerous to living things, but alpha decay is the least dangerous.

Figure 3.2: The first four SciQ training set examples. An instance consists of a question and four answer options (the correct one in green). Most instances come with the document used to formulate the question.

3.4 SciQ: Dataset Properties

Following the previously described data annotation procedure, we collected a total of 13,679 multiple choice questions, which make up the SciQ dataset. This dataset is shuffled into random order and divided into training, development, and test splits, using 1,000 samples for the development and test split each, and the rest as training samples. The initial four samples of this dataset are shown in Figure 3.2.

Each question is naturally associated with the text passage used to write the question about. If this passage were given in the multiple choice setting, then the answer would in many cases be trivial to select as the only one of the candidates mentioned in this given paragraph. For this reason, the multiple choice setting in SciQ comes without the passage; systems trying to answer these questions thus have to find relevant background information elsewhere, e.g. via retrieval.

A different setup – the direct-answer version of SciQ – has the paragraph and question given, yet none of the answer candidates. While most of the questions and texts are made available to the broader research community under a Creative Commons license, a small fraction of paragraphs is withheld due to copyright restrictions. The direct answer setting of SciQ thus has a slightly reduced size with

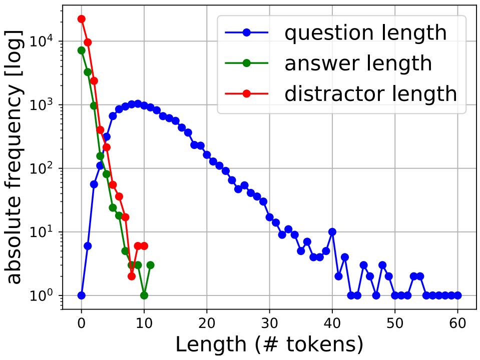


Figure 3.3: Absolute frequency of different lengths for questions, correct answers, and distractors – measured in number of tokens, and calculated across the training set.

only 10,481 questions in the training, 887 in the development, and 884 questions in the test set, as the corresponding paragraphs cannot be freely distributed.

Question and Answer Length In Figure 3.3 we plot empirical distributions for the length of questions and answers in the data; note the log scale of the ordinate. In the majority of cases both the question and answer are relatively short, with only a minority of very long questions. The distributions for the number of tokens in the correct answer and distractors coincide to a large degree; length alone is thus not a salient characteristic to distinguish correct answers from false candidates.

Distinguishing Real from Crowdsourced Questions To obtain an indication for the extent to which questions in SciQ differ from real science exam questions, we set up another annotation task with the following setup: an annotator is given both an original science exam question and a SciQ question, in randomised order (Figure 3.4 shows an example). Annotators are then instructed to predict which of these two questions is more likely to appear as a real exam question. We paired 100 original exam questions with 100 randomly chosen instances from the SciQ training set, and found that annotators were able to identify the actual exam question in 55% of the cases. This indicates that while there are notable differences between original and crowdsourced questions, the plausibility – judged by non-experts – of SciQ questions is broadly comparable to real exam questions.

When inspecting questions in SciQ ourselves, we further observed that especially in cases of multi-word answers, the questions can be distinguished by the

Q: In the life cycle of a fly, which stage comes after the larval stage?
A: **pupa**
B: adult
C: nymph
D: egg

Q: What do metals typically lose to achieve stability?
A: **electrons**
B: ions
C: atoms
D: molecules

Figure 3.4: Comparison of an original exam question (left) with a question in SciQ (right).

structural similarity of their answer candidates. For example, in the question *"What can damage the hair cells lining the cochlea of the inner ear?"* all the answer options follow the same pattern: "*loud sounds*", "*amelodic sounds*", "*unexpected sounds*" and "*wavering sounds*"; this is a consequence of the constrained structure of $\bar{C}$ for multi-token candidates. In contrast, multi-word answer candidates for real exam questions are structurally not as rigid and more varied: for example the question "*Where will a sidewalk feel hottest on a warm, clear day?*" has the answer options "*in direct sunlight*", "*under a picnic table*", "*under a puddle*" and "*in the shade*".

3.5 Experiments

SciQ is intended as an RC dataset, with a particular emphasis on the science exam domain. We will next establish a human baseline and then discuss, first, whether neural RC approaches can learn abilities on the SciQ training questions that generalise to held out test questions; second, how neural RC approaches compare to previously established science exam QA methods; and third, whether SciQ training questions can be used to improve neural RC model performance on real science exam questions.

3.5.1 Human Baseline

To put any results of the subsequently tested models into relative perspective, we first establish a human performance baseline. To this end, we distribute a subset of 650 SciQ questions to 13 researchers familiar with the science exam QA task, and ask them to answer 50 questions. Individuals are not given the supporting SciQ document associated with the question, but instead permission to query the web and no time constraint for selecting the answer. All individuals fully completed this task, and the overall accuracy of these individuals is 87.8% on average, with a standard deviation between the scores of different subjects of 0.045.

It is worth pointing out that a comparison of this observed accuracy value to a corresponding score on real science exam questions is possible only in a limited sense. Real exam questions are used as an assessment for both a class' and individual student's knowledge and learning progress. Following the assessment scale of the real exam questions we consider, the score of – on average – 87.8% indicates "Meeting the Standards with Distinction",[5] but the individuals we tested likely benefited from the lack of a time constraint, and direct web access. When considering the performance of human students on real science exam questions, there is typically also a *distribution* in performance – with substantially larger variance than what we observed. The individuals we tested were furthermore familiar with the science exam QA task and had previously been exposed to several of the real exam questions.

Overall, the relatively high score achieved on SciQ demonstrate that the large majority of questions can be answered correctly by adult individuals with a university education and task familiarity.

3.5.2 SciQ: Multiple Choice Setting

We next consider automatic methods for solving science exam questions. A variety of such models has been developed, notably the *Aristo* model ensemble (Clark et al., 2016) by the Allen Institute for Artificial Intelligence. We will discuss the performance of *Aristo* on SciQ, as well as two of its constitutent sub-components: the *Lucene* information retrieval baseline; and *TableILP*, a table-based integer linear programming model.

The *Lucene* method – which uses IR to search for texts that are relevant to a query and uses search scores for forming an answer prediction – has previously been shown to be a remarkably strong science exam QA baseline (Clark et al., 2016). *TableILP* utilises derivations from a set of knowledge items which have been identified as relevant for the science exam QA task. We are in particular interested

[5] `https://www.nysedregents.org/grade8/science/618/home.html`

Model	**QA Accuracy**
Aristo (Clark et al., 2016)	77.4
TableILP (Clark et al., 2016)	31.8
Lucene (Clark et al., 2016)	80.0
AS Reader (Kadlec et al., 2016)	74.1
GA Reader (Dhingra et al., 2016)	73.8
Human	87.8 ± 0.045

Table 3.2: Test set accuracy of existing models on the multiple choice version of SciQ.

in how they compare to the performance of two off-the shelf neural RC models.

The two neural RC models we consider are the Attention Sum Reader (*AS Reader*; Kadlec et al. (2016)) and the Gated Attention Reader (*GA Reader*; Dhingra et al. (2016)). These RC models can answer multiple-choice questions but require a relevant text passage to read. To this end, the background corpus of *Aristo*'s *Lucene* retrieval model is used to retrieve potentially relevant text passages, which the RC models then process and base their prediction on. Concretely, this follows the approach of Clark et al. (2016) where five IR queries are issued based on either the question text alone, or together with each of the four answer candidate options, concatenated to the question. The top three retrieved results for each of these queries are concatenated to form a larger paragraph, which is read by the RC models. In Table 3.2 we list the resulting model performance on the SciQ test set.

A first observation is that all models – besides *TableILP* – perform relatively well, albeit still with a substantial margin to human performance. The performance of the *Aristo* ensemble on SciQ is comparable but slightly above its accuracy for real science exam questions (where this system achieves 71.3% accuracy (Clark et al., 2016)). Next we observe that with 31.8%, *TableILP* does not perform well compared to the other methods, only 6.8% above a random answer selection baseline, and 12.0% worse than on real exam questions (43.8%; cf. Fig. 2 in Clark et al. (2016)). The method uses a manually curated collection of structured background knowledge items, which is likely only to a limited extent relevant to the new questions in SciQ. This demonstrates a limitation of the *TableILP* method, which performs substantially better on a set of exam questions with topics which its corpus of background knowledge is curated for, but shows a sharp deterioration when tested on the new SciQ questions.

Next, it can be considered surprising that *Lucene* – a subcomponent of the *Aristo* model ensemble – reaches a higher score than the full ensemble. This is due to the fact that neither the trainable components of *TableILP* nor *Aristo* as a whole where retrained for SciQ, but the solvers were taken off-the-shelf and evaluated on the new SciQ test questions.[6] It is worth pointing out that the retrieval corpus of the *Lucene* baseline overlaps with the corpus of study books used to create SciQ. Consequently the performance of *Lucene*, but also of *Aristo* – which uses *Lucene* as a component – as well as the *AS* and *GA readers* that rely on these retrieved documents, is positively impacted by the high retrieval success rates of *Lucene*, and likely larger than without the availability of directly relevant documents to the IR system.

Finally, the two neural RC methods (*AS reader* and *GA reader*) achieve high scores on SciQ, and are almost on par with the *Aristo* ensemble. This is remarkable, since *Aristo* has received a considerable amount of research and engineering through several iterations to succeed on the science exam QA task. Interestingly though the neural RC models perform worse than the *Lucene* information retrieval baseline, even though they use precisely the same retrieved documents. *Lucene* chooses as prediction the option with the highest retrieval score, i.e. the option with the highest (frequency-weighted) degree of lexical overlap between retrieved document words, and each candidate. The strong performance of *Lucene* thus indicates that there is a substantial degree of lexical overlap between SciQ questions and correct answer option (considered together) with the original documents that these questions were written about. Like *Lucene*, the two neural models rely on successful retrieval of relevant documents (as prior step). But unlike *Lucene*, the two neural models do not have direct access to the predictive retrieval score used by *Lucene*, and achieve slightly lower, albeit overall comparative QA performance.

[6] Note that this also includes the relative weighting of model contributions in the *Aristo* ensemble prediction.

3.5.3 Using SciQ to Answer Exam Questions

The previous section has shown that it is possible to successfully tune neural RC models which can generalise their answering skills to new, held out science questions within SciQ. But both the training and test questions in SciQ are only an approximation of real exam questions and written in a crowdsourced process by non-experts. The next question, then, is whether questions assembled in SciQ can be used to improve an RC system's accuracy on real exam questions. To this end, an experiment was conducted on science exam questions from both 4th and 8th grade.[7]

The *AS* and *GA reader* were trained, first, only on real exam questions, and then, in a separate experiment, on exam questions together with SciQ, separately both for 4th and 8th grade questions. Table 3.3 shows model accuracies on these real exam test questions with and without the augmentation by SciQ training questions. We observe that adding SciQ results in accuracy improvements for both of the RC models, and for both of the grade levels. This confirms that SciQ questions can provide a useful training signal to help neural RC systems learn reading skills that can be applied to solve science exam questions. Interestingly, the improvements are larger on the 4th grade questions, where there are fewer real training questions than for the 8th grade level.

Overall it is however worth pointing out that these accuracy rates lack behind those reported on the SciQ test set (see Table 3.2). We have already established that the overlap of the retrieval corpus with the texts underlying the SciQ questions provides high levels of retrieval success, as visible in the high scores of the *Lucene* baseline. Real exam questions, on the other hand, are not composed based on directly relevant and retrievable text passages – as was done in the assembly of SciQ. Consequently the QA performance on real questions is sharply reduced in comparison, as there are not as many directly relevant documents available in the underlying corpus.

[7] There are approx. 3,200 8th grade questions and 1,200 4th grade questions. Some of the questions come from `www.allenai.org/data`, others are proprietary.

Dataset	**AS Reader**	**GA Reader**
4th grade	40.7%	37.6%
4th grade + SciQ	45.0%	45.4%
Difference	+4.3%	+7.8%
8th grade	41.2%	41.0%
8th grade + SciQ	43.0%	44.3%
Difference	+1.8%	+3.3%

Table 3.3: Including SciQ samples during model training increases neural RC model accuracy on real science exam questions, both for 4th and 8th grade exam questions.

3.5.4 SciQ: Direct Answer Setting

The previously described multiple-choice experiments consider the RC components as part of a QA pipeline, following an information retrieval step. We will next consider how well a neural RC model can perform on the *direct answer* setting in SciQ. Here, no answer candidates are given, but instead the document which the question was written about – effectively giving the QA model oracle access to a relevant document. We consider the Bidirectional Attention Flow model (*BiDAF*) (Seo et al., 2017a), a widely used model for extractive RC. On the SciQ test set *BiDAF* achieves exact match and F_1 scores of 66.7 and 75.7, respectively. This is only 1.3% and 1.6% below the corresponding values achieved on SQuAD1.1, and these numbers are broadly comparable to those reported for the neural RC models in Table 3.2.

3.6 Discussion and Conclusion

We have observed that neural RC systems can successfully be applied on SciQ and learn to generalise question answering behaviour to held out test questions, both when coupled with a document retriever and when oracle documents are given. The evaluation on the specialised domain of science exam study material provides additional support for these RC models and validates their capability to learn generalisable RC skills.

We have furthermore identified access to relevant background information as a critical factor for model performance in a science exam QA system: while *TableILP* lacks relevant background knowledge items that apply to SciQ questions, neural RC approaches given relevant documents (either through retrieval, or by an oracle) perform relatively well. This is in line with observations in open-domain QA (Chen et al., 2017a) and Fact Checking (Thorne et al., 2018), where the most critical bottleneck to overall system performance appears to be the availibility of relevant textual information as provided by the retrieval component in a system pipeline. Finally, we have seen that SciQ is a useful ingredient to support RC systems when solving real science exam questions.

3.6.1 General Dataset Limitations

Besides the overall encouraging experimental results, it is worth pointing out several limitations of the SciQ dataset.

First, as we follow a crowdsourcing approach in which annotators are given a text passage to formulate a question about, the resulting questions have a considerable degree of lexical overlap with the given passage. This is reflected in the strong performance of the IR baseline, which – based on lexical overlap – reaches a higher score than the two neural models (see Table 3.2). Given the prevalence of this predictive signal in the data it is unclear to what extent RC models trained on SciQ transfer to settings with less lexical overlap. We point out that subsequent work has found ways to address this problem in their dataset construction approach, albeit in another domain (Kočiský et al., 2018).

Second, SciQ is focused on relatively short text segments. While partly a response to memory limitations of many RC models, this setup biases the resulting dataset towards textual content which can be explained in a relatively short context – thus limiting, for example, the degree of co-reference and total amount of information in the given pieces of text.

Third, an inherent limitation of using a model to suggest alternative answer candidates is that they will implicitly reflect properties of this model – its features, training data points, and modelling limitations. While this is true in general, one particular limitation is the one we have observed for multi-word answer candidates: our predictive model often presents plausible alternatives for single-word answers, but its suggestions on candidates with more tokens are often less plausible. Future work using generative models may further improve multi-word candidate suggestions by beginning with more relevant expressions to be scored by the ranker than the set $\bar{C}$ we chose, which is very limited for multi-token candidates.

A final – and important – consideration is that the SciQ question distribution is one of comprehension questions written about a given piece of in-domain text. In this regard the dataset differs from real exam questions which are typically posed without such a concrete available text source. That is, only for few real exam questions there is a piece of relevant text that explicitly states this requested information, but in the general case this cannot be expected. This brings the real exam question setting much closer to that of web search engine or QA system questions (Nguyen et al., 2016; Kwiatkowski et al., 2019). This mismatch is also reflected in the types of questions we find in SciQ: they tend to request factual information about the given passages, which is often stated explicitly, rather than involve more complex derivations or inference in hypothetical situations – which is more often found in real exam questions.

We thus see SciQ as a potentially useful contributory data resource which may help an algorithmic science exam solver learn text comprehension behaviour. It is however not intended to be a standalone dataset which is by itself sufficient – even if further scaled – to comprehensively learn all the relevant skills necessary to answer exam questions.

3.6.2 Summary of Answers to Initial Research Questions

Finally, we summarise the answers to our initial research questions posed in the chapter introduction:

1. **How can the crowdsourcing approach to RC dataset creation be adapted for collecting a dataset in the science exam domain?** We have described a two-stage annotation approach, in which the first stage involves writing a question about a paragraph from a corpus of digitally available study text books, followed by a second step in which annotators both validate the previously written questions and add false multiple-choice answer candidates,

supported by model suggestions.

2. **How do previously developed science exam solvers and neural RC methods perform on the resulting dataset?** The previously used *Aristo* ensemble performs similar on SCIQ compared to real exam questions; *TableILP* performance on the other hand drops, likely due to a lack of relevant background knowledge items for these new questions. The *Lucene* baseline reaches a higher score than on real exam questions, likely due to having access to the corpus of directly relevant documents that the question were written about, and lexical overlap of these articles with the questions. When coupled with IR outputs, neural RC models achieve comparable performance to *Aristo* and *Lucene* in the multiple-choice setting, and only a small performance deterioration in the direct-answer setting, indicating that the availability of relevant documents to read is the main bottleneck for QA performance of neural RC systems.

3. **Can the resulting data be used as additional training resource to improve the performance of RC systems on science exam questions?** We observe that adding the SCIQ dataset to a smaller training dataset of real exam questions boosts the performance of two neural RC models, both for 4th and 8th grade questions.

Part II

Multi-Hop Machine Comprehension

Chapter 4

Constructing Datasets for Multi-hop Reading Comprehension across Documents

> *The content of this chapter is based on previously published work (Welbl et al., 2018). The chapter includes descriptions for the construction of two datasets, following a shared methodology. The assembly of the* MEDHOP *dataset and its annotation were conducted by a coauthor in Welbl et al. (2018).*

The maturation of end-to-end RC methods has led to systems that can learn to identify correct answers to text comprehension questions at a level that has approached and surpassed human-level performance on SQUAD (Kadlec et al., 2016; Seo et al., 2017a; Yu et al., 2018). However, the focus of SQUAD is on questions about a single document, and within this given document the information necessary for answering the comprehension question is often very locally concentrated: prior work (Min et al., 2018) found that for 90% of samples in SQUAD only a single sentence is relevant to answer the given question. Furthermore, relevant sentences in SQUAD typically exhibit a substantial degree of word overlap with the comprehension question, which makes lexical overlap an informative cue for selecting the correct answer. For example, Weissenborn et al. (2017) demonstrated that using a binary *word-in-question* indicator feature improves the relative accuracy of a

Document 1: The Hanging Gardens, in Mumbai, also known as Pherozeshah Mehta Gardens, are terraced gardens […] They provide sunset views over the Arabian Sea […]

Document 2: Mumbai (also known as Bombay, the official name until 1995) is the capital city of the Indian state of Maharashtra. It is the most populous city in **India** […]

Document 3: The Arabian Sea is a region of the northern Indian Ocean bounded on the north by **Pakistan** and **Iran**, on the west by northeastern **Somalia** and the Arabian Peninsula, and on the east by **India** […]

Query: (Hanging Gardens of Mumbai, country, ?)
Answer Options: {**India**, **Pakistan**, **Iran**, **Somalia**}

Figure 4.1: An example from the WikiHop dataset where, to infer the correct answer, it is necessary to combine information spread across multiple documents. Underlined entities of different colours in Document 1 appear also in other contexts (Document 2 and Document 3), together with the correct answer (green) and false answer options (grey).

baseline model by 27.9%.

A possible explanation for these observations is the crowdsourcing approach to comprehension question writing: annotators have an incentive to complete the annotation task as quickly as possible. This is most easily achieved by following the least cognitively demanding route, and often results in questions about explicitly stated information in the given text, i.e. its base interpretation. As a consequence, the resulting models trained on this data emphasise the role of locating, matching, and aligning relevant words between comprehension question and given text, rather than developing more general and abstractive text comprehension and inference capabilities.

One direction for advancing the abilities of machine comprehension methods, then, is to progress towards reading scenarios where the information relevant to the comprehension question is not explicitly stated within a single sentence or document, and where the answer cannot be inferred directly. Consider for example Figure 4.1, where a query about the `country` property of the *Hanging Gardens of Mumbai* is given, together with three different Wikipedia articles. The correct answer to the `country` property (*India*) cannot be inferred by reading any one of the articles alone without additional background knowledge; the answer is not stated explicitly in the article about the *Hanging Gardens* itself. However the linked articles both mention the correct answer *India* (as well as other countries), in connection with entities appearing in the same article as the *Hanging Gardens* (*Mumbai* and the *Arabian Sea*).

Finding the answer in this example requires *multi-hop* inference: first, inferring that the *Hanging Gardens* are located in *Mumbai*; second, and from a separate document, that *Mumbai* is a city in *India*; this together entails that the `country` property of the *Hanging Gardens* is *India*. Text comprehension thus involves the integration of information from a context spanning several documents, thus reaching beyond what is typically required to answer questions in SQuAD, or the SciQ dataset from the previous chapter.

Extending the scope of text comprehension methods with the ability to integrate textual information across various documents could aid applications of Information Extraction (IE), such as discovering drug-drug interactions (Gurulingappa et al., 2012) by connecting protein interactions reported in different scientific publications. It could also benefit text search (Carpineto and Romano, 2012) and QA applications (Lin and Pantel, 2001) in which required information is not always comprehensively and explicitly stated in a single sentence or document. However, the progress and development of RC methods with cross-document multi-hop inference abilities has in the past been impeded by a lack of large-scale dataset resources.

In this chapter we will define an RC task in which a model has to learn to answer queries by reading and combining textual evidence stated in multiple documents. Modelling progress generally depends on the availability of datasets, both for training and for evaluation. We will thus initially introduce a dataset induction methodology for this task, which we then apply to construct two datasets. The first of these datasets, WikiHop, contains sets of Wikipedia paragraphs where answers to queries about particular attributes of a given entity cannot be located as a span in the paragraph associated with the entity's Wikipedia article, like in the example given in Figure 4.1. For the second dataset, MedHop, the objective is to predict interactions between drug pairs based on information stated in combinations of Medline research paper abstracts which cover scientific findings about drugs, proteins and interactions between them. During the construction of the two datasets we utilise existing Knowledge Bases (KBs) – WIKIDATA and DRUGBANK– as ground truth, and follow prior work (Hewlett et al., 2016; Joshi et al., 2017) in utilising distant supervision (Mintz et al., 2009).

We will observe that assembling datasets for cross-document RC poses challenges: there exists a variety of pitfalls that can render the resulting datasets trivial to solve using shallow statistical predictors, e.g. spurious co-occurrences of answers with particular documents. Thus, woven into the dataset induction description there will be a thread of remedial procedures that address and mitigate these issues.

After describing the dataset construction in detail we will establish a human performance baseline, and analyse to which extent assumptions made during dataset assembly were justified. We then compare several baseline methods on the two resulting datasets – both shallow statistical predictors and established neural RC architectures – analysing various aspects of their behaviour in ablation studies. In summary, this chapter aims to answer the following research questions:

List of Research Questions Addressed in this Chapter:

1. How can we construct datasets for multi-hop Reading Comprehension across documents?

2. What are potential dataset biases and pitfalls associated with the dataset assembly approach chosen?

3. How do two neural RC models – *FastQA* and *BiDAF* – perform on the resulting cross-document multi-hop RC datasets, in particular compared to shallow statistical baselines?

4.1 Dataset Induction Method

4.1.1 Task Formalisation

We will begin by defining the multi-hop RC task more formally and then describe a general method for dataset induction. In Sections 4.2 and 4.3 we will show how the methodology can be applied, and describe the creation of such datasets in two domains.

In the multi-hop RC task, a model is in each sample given a query q (either posed in natural language, or structured; we focus on the latter case), a set of supporting documents S, as well as a set of candidate answers C. We assume that all answer candidates are mentioned in S. The goal for the model is to predict the correct answer $a^* \in C$ based on both q and S.

Generally, queries can potentially have several true answers if they are not constrained to a specific set of given documents, for example a query about the parent of a particular person. In our setup the samples are however intended to only have one correct answer among the given candidates C based on S.

It is further worth pointing out that although we will make use of background information when assembling the datasets, such information is generally not available to the model: the set of documents is given in randomised order and without further metadata (such as hyperlinks) or additional information about the documents. While we expect such information to potentially be beneficial, this would likely not be available in the same form in different domains, and distract from our goal of encouraging RC methods to infer new facts by combining information stated in separate texts.

4.1.2 Dataset Induction Using a Bipartite Graph

In this section we will describe an automatic dataset induction strategy with which we assemble a collection of samples in the format laid out above. The method requires a document corpus D, as well as a KB of fact tuples (s, r, o), where s is a subject entity, r a relation type, and o an object entity. One example of such a tuple is `(Hanging_Gardens_of_Mumbai, country, India)`.

In our first dataset construction step we begin with a set of such KB fact tuples. We convert them individually into query-answer pairs by removing the object slot, i.e. $q = (s, r, ?)$, and then using the object as the correct answer, i.e. $a^* = o$.

We then introduce a directed bipartite graph, in which one side of vertices corresponds to individual documents in D, and the other side of vertices to particular

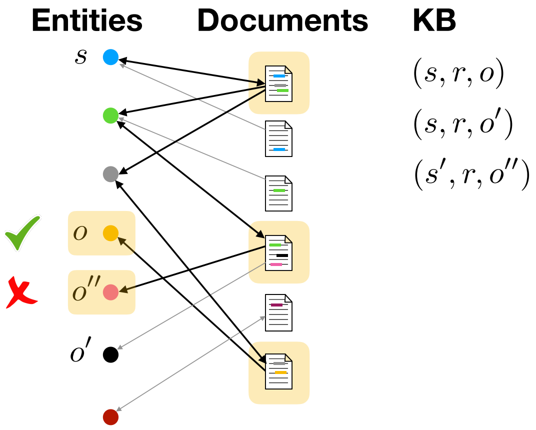


Figure 4.2: A bipartite graph connecting entities and documents mentioning them. Bold edges are those traversed for the first fact in the small KB on the right; yellow highlighting indicates documents in S and candidates in C. Check and cross indicate correct and false candidates.

KB entities; Figure 4.2 shows an example. In this graph, a node d (corresponding to a document) is connected to a node e (corresponding to an entity) if e is mentioned in d, though there may be additional constraints on the connectivity of the graph. To identify possible support documents and answer candidates for a given (q, a^*) pair, this bipartite graph is then traversed with a breadth-first search – beginning at the node of the subject entity s of q. The possible end points of the graph traversal are chosen as the set of all entity nodes which are type-consistent answer entities to q (more on that below).

The graph traversal can – beginning from s – potentially visit several end points, although generally not necessarily all. End points which are visited will become the set of answer candidates C for q, which is usually substantially smaller than the set of *all* potential type-consistent answers to q. We discard (q, a^*) pairs in which a^* is not among the end points visited. For the remaining samples, the documents visited along the paths towards end points which *were* reached define the set S of support documents for q. S thus comprises document chains connected by entities mentioned in both of two linked documents, and which lead not only from s to the correct answer, but also towards documents mentioning false other type-consistent answer candidates.

In order to identify a set of type-consistent answer entities to the query q in the first place, all entities in the KB are considered. Those which appear as the object in at least one fact with r as relation type are considered type consistent and thus potential candidates – notably also including a^*. If another fact (s, r, o') exists in the KB, i.e. a fact indicating another true answer to the same query, then we exclude o' from the set of graph traversal end points for this particular sample. Thus, relying on a closed-world assumption for the KB, only one of the end points resembles a correct answer to the query.[1]

Following this general data induction approach, we will consider two concrete applications: first in the case of WIKIPEDIA, and second in the case of PUBMED, each with its own intricacies.

4.1.3 Discussion

By using the methodology described above, the query entity s and correct answer a^* are located in separate documents, linked by a chain of intermediate other documents and entities they coappear with. This means that potentially relevant textual evidence for (q, a^*) is spread across the chain of documents linking s with a^*. In this manner, multi-hop inference requires more than co-reference resolution in one of the given documents alone.

It is worth pointing out that the inclusion of type-consistent alternative candidates besides a^* results in an additional challenge for a model. False answer candidates counterbalance potential predictive regularities among the answers: models could otherwise predict a^* based on their type (Jia and Liang, 2017; Lewis and Fan, 2019). For example, if only a single country is mentioned in the given documents, it is much easier for a model to identify this country as the correct answer to a `country` query – without having to take into account the query subject, or other information from the text. By introducing multiple answer candidates (and corresponding documents), our dataset assembly method stands in contrast to prior work, which avoids such cues with masking (Hermann et al., 2015; Hill et al., 2016). While entity masking removes type information, it also masks out potentially relevant lexical information; in the Experiments Section 4.6 we will analyse the potential impact of

[1] That is, we assume that the facts in the KB state all true facts.

this in the context of our resulting datasets.

One shortcoming of the dataset induction strategy is its dependence on previously identified entities, as well as a given KB. Both can be sources of erroneous or noisy samples, and we will thus later investigate the extent of this problem in a qualitative analysis of the resulting data points. A second shortcoming is the reliance on the closed-world assumption: if facts do not appear in the KB we assume them to be false, leading to potentially false negative answer candidates where the closed-world assumption is violated. Next, the task we pose relies on structured queries of a particular format. While this allows for more control regarding the requested information, it does not cover the full variation of possible comprehension questions that could be posed using *natural* language. Furthermore we prescribe an *extractive* format for our RC task, similar to SQuAD. While both these latter factors impose significant restrictions, it facilitates automatic assembly and evaluation, and the extractive character of the task allows for easier transfer of previously developed RC approaches to the cross-document RC task we have introduced.

The key advantage of the method is its fully automatic character: it does not rely on humans in the loop, although we rely on human annotations for validation. It is thus more cost-efficient and easily scalable where KB and corpora are available – an important factor to train parameter-rich neural RC models. We furthermore circumvent dataset annotation biases introduced by crowdworkers, and queries are not posed conditioned on a given text; instead the text is selected based on the query and its answer. However, as we rely on the distant supervision assumption (Mintz et al., 2009), there is no guarantee that the provided documents actually serve as valid support for inferring the correct answer. In our view, this poses perhaps the most significant limitation to the approach; to what extent the assumption is justified will thus also be analysed further below, after describing the creation of both datasets.

4.2 Dataset Induction: WIKIHOP

WIKIPEDIA presents a large corpus of comparatively clean encyclopedic text from a variety of domains, and it is associated with structured knowledge resources, e.g. WIKIDATA (Vrandečić, 2012). WIKIPEDIA is a resource widely used in RC research, although mostly for datasets with queries about individual sentences (Morales et al., 2016; Levy et al., 2017) or articles (Yang et al., 2015; Hewlett et al., 2016), including SQUAD (Rajpurkar et al., 2016). Prior to this study, no attempts were undertaken to assemble a multi-step RC dataset involving multiple WIKIPEDIA documents.

A dataset closely linked to the one we will assemble is the WIKIREADING (Hewlett et al., 2016) dataset. It is based on WIKIPEDIA and uses WIKIDATA tuples of the form `(item, property, answer)` which are matched with WIKIPEDIA articles associated with their `item`. In WIKIREADING these tuples are used for a slot filling task, which consists in identifying the `answer`, given an `article` and `property` – thus resembling the same type of query also used in our dataset induction strategy. But one issue with WIKIREADING is that 54.4% of samples do not mention the correct answer explicitly in the given text (Hewlett et al., 2016), limiting its usefulness as an extractive RC dataset. We observed, however, that several other WIKIPEDIA articles accessible through hyperlinks in the given article frequently mention the answer, as well as other plausible alternative candidates. This has inspired our dataset construction method, and we will use WIKIREADING as a starting point to develop our own dataset: WIKIHOP.

4.2.1 Assembly

We next apply the method described in Section 4.1.2 to assemble a multi-hop dataset using WIKIPEDIA as corpus and WIKIDATA as KB of fact triples. That is, the `(item, property, answer)` WIKIDATA triples correspond to (s,r,o) tuples; and the `item` and `property` of each sample together form the query q – for example *(Hanging Gardens of Mumbai, country, ?)*. Following prior work (Yang et al., 2015) we utilise only the first paragraph of a given article, as relevant information is frequently stated at the start.

Beginning with the full WIKIREADING dataset, we delete samples in which the answer is mentioned explicitly in the given WIKIPEDIA article about the item.[2] The bipartite graph then has the following connectivity structure: (1) for edges from articles to entities: all articles mentioning an entity *e* are connected to *e*; (2) for edges from entities to articles: each entity *e* is only connected to the WIKIPEDIA article about the entity. Traversing this graph is thus equivalent to iteratively following hyperlinks to new articles about anchor text entities.

Given a WIKIDATA query-answer pair, the item entity forms the graph traversal starting point. The traversal will thus always visit the article about the item, as it is the only document connected from there. The set of end points comprises both the correct answer and alternative type-consistent candidate expressions. They are selected based on all facts appearing in the WIKIREADING training set, choosing tuples with the property of q, and collecting the answer expressions of these facts. For example, the country property in WIKIDATA has this set of type-consistent entities (and thus potential answer candidates): {*Italy*, *United Kingdom*, *...*}.

Overall, graph traversal is carried out to a maximum of three documents per document chain. In order not to impose unnecessary computational burden, we remove samples exceeding 64 different support documents, or 100 candidates, which amounts to $\approx 1\%$ of the examples.

4.2.2 Mitigating Dataset Biases

The creation of a new dataset is prone to the unintentional introduction of biases (Chen et al., 2016; Schwartz et al., 2017). As the analysis of the WIKIREADING dataset was limited (Hewlett et al., 2016), we now demonstrate some of the downstream effects observed on WIKIHOP.

Candidate Frequency Imbalance A first finding is that there exists a considerable bias in the distribution of answers in WIKIREADING. For example, the majority of examples with the country property has as correct answer the *United States of America*, which is due to the geographical topic coverage of English WIKIPEDIA articles. The extent of this imbalance is so substantial that after performing our graph-

[2] We thus use a disjoint subset of WIKIREADING in comparison to Levy et al. (2017).

traversal procedure, 47.8% of the resulting `country` queries have *United States of America* as the correct answer. Even more, since the `country` query type tends to be very prominent, 20.8% of *all* samples in the entire dataset have this answer. Clearly a simple baseline which always predicts the majority class (perhaps conditioned on the query type) would prove already moderately successful, yet without demonstrating any multi-hop comprehension. To address the issue we thus sub-sample our dataset such that no more than 0.1% of the examples share the same answer.

Document-Answer Correlations A second problem – one unique to the cross-document task setup – is the potential for spurious correlations between answer candidates and particular documents, which can result from the connectivity pattern of the dataset construction graph. We found that specific documents frequently co-occur with the correct answer in the same data sample, while others do not. If the WIKIPEDIA article about *London*, for example, appears in S, then the correct answer is very likely the *United Kingdom*; this can be determined without consideration of the query type or query entity.

In order to quantify the extent of this problem we design a statistical metric that measures the effect, which we then use again to sub-sample the dataset. This metric counts the number of training samples in which a candidate c is the correct answer when a particular document d is present among S. That is, for a given document d and candidate c, the metric $cooccurrence(d,c)$ denotes the total number of co-occurrences of d with c in the same sample, and where c is the correct answer. Table 4.1 shows a ranked list of document-answer pairs that most frequently co-appear in this way. Since the mere presence of one of these documents in S can resemble a very strong cue for the correct answer, a model could thus solve the task without having to rely on RC capabilities at all.

To address this problem we utilise the above introduced co-occurrence metric to further sub-sample the WIKIHOP dataset, removing examples for which one or more pairs (d,c) of document and candidate have $cooccurrence(d,c) > 20$. That is, we remove data points containing a document and candidate which frequently occur together, and where the candidate is the correct answer. Before and after

Answer a^*	**WIKIPEDIA article** d	**Count**	**Prop.**
united states of america	A **U.S. state** is a constituent political entity of the United States of America.	68,233	12.9%
united kingdom	**England** is a country that is part of the United Kingdom.	54,005	10.2%
taxon	In biology, a **species** (abbreviated sp., with the plural form species abbreviated spp.) is the basic unit of biological classification and a taxonomic rank.	40,141	7.6%
taxon	A **genus** (pl. **genera**) is a taxonomic rank used in the biological classification	38,466	7.3%
united kingdom	The **United Kingdom of Great Britain and Northern Ireland**, commonly known as the **United Kingdom (UK)** or Britain, is a sovereign country in western Europe.	31,071	5.9%
taxon	**Biology** is a natural science concerned with the study of life and living organisms, including their structure, function, growth, evolution, distribution, identification and taxonomy.	27,609	5.2%
united kingdom	**Scotland** [...] is a country that is part of the United Kingdom and covers the northern third of the island of Great Britain.	25,456	4.8%
united kingdom	**Wales** [...] is a country that is part of the United Kingdom and the island of Great Britain.	21,961	4.2%
united kingdom	**London** [...] is the capital and most populous city of England and the United Kingdom, as well as the most populous city proper in the European Union.	21,920	4.2%
united states of america	**Nevada** (Spanish for ”snowy”; see pronunciations) is a state in the Western, Mountain West, and Southwestern regions of the United States of America.	18,215	3.4%
...	...	...	
italy	The **comune** [...] is a basic administrative division in Italy, roughly equivalent to a township or municipality.	8,785	1.7%
...	...	...	
human settlement	A **town** is a human settlement larger than a village but smaller than a city.	5,092	1.0%
...	...	...	
people’s republic of china	**Shanghai** [...] often abbreviated as Hu or Shen, is one of the four direct-controlled municipalities of the People’s Republic of China.	3,628	0.7%

Table 4.1: Pairs of correct answer and article, sorted by their $cooccurrence(d,a^*)$ statistic (before any filtering; total size: 527,773). The *Count* column states the value of $cooccurrence(d,a^*)$; the last column states the corresponding total proportion of samples in the training set.

sub-sampling, we measure how easy it is for a model to exploit these potentially informative document-answer co-occurrences. Concretely, we define the following statistical prediction model which selects the candidate with highest *cooccurrence* score across the given documents:

$$\arg\max_{c \in C} [\max_{d \in S}(cooccurrence(d,c))]$$

In fact, before sub-sampling, this shallow statistical baseline model achieves 74.6% accuracy on the task (more details on this in Section 4.6). This is a surprisingly strong result, and one can see how it could distort our interpretations of multi-hop capabilities for a neural RC model when tested on this dataset, even though it might just have learned to exploit this regularity. After sub-sampling, the performance of this predictive statistic drops to 36.7% accuracy. That is, this statistic still constitutes a strong baseline model, but does not stand out as the single predominant signal for predicting the answer any more.

This concludes our description of the assembly procedure for creating the WIKIHOP dataset. We will next describe how the same general method described in Section 4.1.2 is applied to create a different dataset in another domain: MEDHOP; subsequently we will discuss properties and experiments for both datasets.

4.3 Dataset Induction: MEDHOP

Next we describe how the previously introduced dataset induction method can be used for the construction of a second dataset in the biomedical domain; the particular task we will focus on is the detection of Drug-Drug Interactions (DDIs). Such interactions between drug pairs are caused by protein-protein interaction (PPI) chains; a pair of drugs can interact if the proteins they target interact with one another. Due to the compositional nature as sequence of interactions, the DDI task lends itself well to our dataset assembly method.

Prior work on detecting DDI relationships from text focuses on explicit interactions stated within a single sentence (Gurulingappa et al., 2012; Percha et al., 2012; Segura-Bedmar et al., 2013). But information about the target proteins of

Document 1: Leuprolide [...] elicited a long-lasting potentiation of excitatory postsynaptic currents[...] GnRH receptor-induced synaptic potentiation was blocked [...] by Progonadoliberin-1, a specific GnRH receptor antagonist [...]

Document 2: [...] our research to study the distribution, co-localization of **Urofollitropin** and its receptor[,] and co-localization of **Urofollitropin** and GnRH receptor [...]

Document 3: Analyses of gene expression demonstrated a dynamic response to the Progonadoliberin-1 superagonist **Triptorelin**. [...]

Query: (Leuprolide, interacts_with, ?)
Answer Options: {**Triptorelin**, **Urofollitropin**}

Figure 4.3: A sample from the MEDHOP dataset; we aim to collect such samples with the dataset induction strategy lied out in this section. Pink and blue spans are proteins, grey and green text represents the false and correct answer options, respectively.

particular drugs, and information about protein-protein interactions can be stated in separate sentences, or even separate published articles. Consider, for example, the set of documents in Figure 4.3, a truncated sample from the MEDHOP dataset. The first document describes that `Leuprolide` (a drug) elicited `GnRH receptor`-induced synaptic potentiations, which are blocked by the protein `Progonadoliberin-1`. The third document states that `Triptorelin` (a different drug) is a superagonist of the same protein (`Progonadoliberin-1`). `Triptorelin` might thus influence the effect of the drug `Leuprolide`, and this is indeed recorded in DRUGBANK. Note that besides this true drug-drug interaction there is also an alternative drug candidate `Urofollitropin`. This drug is mentioned in the same document as another relevant protein, `GnRH receptor`, but Document 2 does not indicate an interaction with the drug `Leuprolide`, thus rendering it a plausible but false answer distractor in this particular sample.

This example illustrates that detecting DDI interactions across document boundaries poses a very challenging scenario, both to human non-experts and to RC models. Mature models for this task could however in the future improve the recall of automatically detected DDIs from the available literature by extending the task across document boundaries. This is in particular relevant considering the composite nature of the DDI relationship, which is mediated via interacting target proteins; reading about it can thus be characterised as a multi-step inference problem. Mature multi-hop methods could help to find and combine individual relevant facts and suggest previously unobserved DDIs, which may not explicitly be described in a given document, but can nevertheless be potentially inferred from several sources.

4.3.1 Assembly

MEDHOP is assembled with DRUGBANK (Law et al., 2014) as the KB of knowledge triples, and a collection of MEDLINE research paper abstracts as document corpus. Among the DRUGBANK facts we only consider those with one type of relationship: `interacts_with`, which connects pairs of drug entities. A concrete MEDHOP query q could thus, for example, be `(Leuprolide, interacts_with, ?)`, with the correct answer a^* `Triptorelin`.

In the bipartite dataset assembly graph only drugs and proteins are considered as the entity nodes. Concretely, the set of entities is limited to drugs in DRUGBANK and human proteins recorded in SWISS-PROT (Bairoch et al., 2004). The document nodes correspond to research paper abstracts from the 2016 MEDLINE release, pre-processed using the 2011 BioNLP Shared Task (Stenetorp et al., 2011) preprocessing pipeline. In summary, the bipartitite graph then has a collection of known drug and protein nodes on one side, and nodes corresponding to MEDLINE paper abstracts on the other.

The connectivity of the bipartite graph follows the same broad principles as for WIKIHOP, albeit with a number of refinements. When linking entities and documents, any name variant of a drug or human protein known in DRUGBANK and SWISS-PROT is used to detect a mention. Similar to prior work (Percha et al., 2012), different name variants of the same drug or protein are normalised as recorded in these databases. The connectivity structure of the graph is then defined as follows: (1) There exists a bidirectional edge between a document and a drug if this document mentions both the drug *and* mentions a protein known to be a target for the drug, according to DRUGBANK. (2) There is an edge from a document to all proteins mentioned within it. (3) There is an edge from a protein p to a document mentioning p, but only if the document also mentions another protein p' that is recorded in REACTOME (Fabregat et al., 2016) as interacting with p. That is, the graph connectivity is restricted to focus on protein pairs known to interact, and proteins known to be the targets of particular drugs. Recall that our dataset induction approach relies on distant supervision, which is a potentially noisy signal. Imposing the above described additional requirements on graph connectivity increases the relevancy of the resulting samples by leveraging additionally available information about the given entities, thus erring on the side of precision.

For a given DRUGBANK fact $(\texttt{drug}_1, \texttt{interacts\_with}, \texttt{drug}_2)$ the subject entity $\texttt{drug}_1$ serves as the graph traversal starting point. The set of possible end points consists of any other drug, but excluding both $\texttt{drug}_1$ and other drugs known to interact with $\texttt{drug}_1$, in order to avoid false negative answer candidates.

Document Sub-sampling Data samples with very large sets of support documents S can impose significant computational challenges for existing neural RC models. As with WIKIHOP, samples are thus limited to a maximum of 64 support documents, and documents restricted to have no more than 300 tokens (plus title). But whereas only a negligible fraction of samples had exceeded 64 documents in WIKIHOP, the bipartite graph for MEDHOP is very densely connected, resulting in the possibility of impractically large sets of support documents S. Thus, following the traversal procedure, document sets are sub-sampled as follows: i) a chain of documents connecting the drug in the query with the correct answer is added; ii) new document chains leading to other answer candidates are added, gradually until a limit of 64 documents is reached, or as many relevant documents as available otherwise. This is carried out in such a way that the different candidates have the same number of paths through the bipartite graph, hence avoiding frequency imbalances in the resulting data samples.

Mitigating Label Imbalance As for WIKIHOP, the resulting dataset is potentially prone to biases, which enables the correct prediction of answers using shallow statistical heuristics. More concretely, some drugs enjoy a higher coverage rate than others, or possess more recorded interaction relationships with other drugs. For

	Train	**Dev**	**Test**	**Total**
WikiHop	43,738	5,129	2,451	51,318
MedHop	1,620	342	546	2,508

Table 4.2: Dataset sizes for our respective datasets.

example *Aspirin* interacts with 743 drugs, whereas there are only 34 interaction records for *Isotretinoin*. Such candidate frequency imbalance issues are problematic, yet due to its smaller total number of samples, sub-sampling MedHop would result in a dataset of insufficient size for the application of neural RC methods. We can nevertheless address this problem through randomly anonymising drug names, which will be explained in more detail in Section 4.6.4.

4.4 Dataset Properties

After having assembled both WikiHop and MedHop, we will next describe some of their basic properties. We will then conduct qualitative analyses and investigate the extent to which some of the assumptions made in our dataset induction strategy are justified.

Dataset Size An overview of the different dataset splits and their size can be found in Table 4.2. Note that WikiHop follows the training, validation, and test dataset splits from WikiReading: the full dataset assembly and sub-sampling pipeline is carried out separately on each of these parts. Sub-sampling to remove frequency biases, and document-answer correlations, reduces the size of WikiHop considerably, from ≈528K samples to ≈44K in the training set. This constitutes a very aggressive dataset size reduction; we opted for this choice to mitigate the dataset biases previously identified, while retaining a large enough dataset to train highly-parameterised neural RC models. MedHop on the other hand is – even without any sub-sampling – a relatively small RC dataset.

Candidate Statistics In Table 4.3 we summarise other general quantitative characteristics of the datasets, including quantities describing the distribution of candidates per sample. Most samples in MedHop have nine candidates, which reflects how document chains in the dataset construction graph are added up to at most 64

	min	max	avg	median
WikiHop: # candidates	2	79	19.8	14
WikiHop: # documents	3	63	13.7	11
WikiHop: # tokens per document	4	2,046	100.4	91
MedHop: # candidates	2	9	8.9	9
MedHop: # documents	5	64	36.4	29
MedHop: # tokens per document	5	458	253.9	264

Table 4.3: Candidates and documents per sample and document length statistics for both the WikiHop and MedHop dataset.

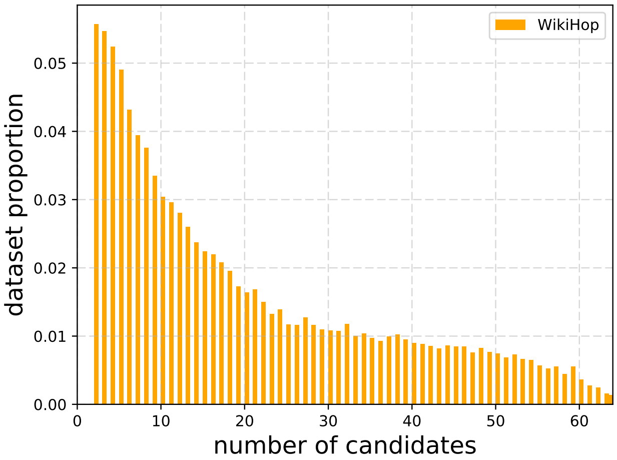


Figure 4.4: Histogram for the number of candidates per sample in WikiHop.

documents. For WikiHop the distribution is more varied, and many queries have a considerable number of answer candidates, with a median of 14. Figure 4.4 further shows a histogram for the distribution of the number of candidates per sample in WikiHop. The number of candidates begins at the lower limit of 2 candidates and is skewed to the left; half of the samples have more than 14 candidates (median).

Document Statistics Besides candidate information, Table 4.3 also lists statistics on the distribution of number of documents per sample, and number of tokens per document. We observe that MedHop has on average a larger number of documents than WikiHop, reflecting its denser dataset assembly graph connectivity. This shift towards larger numbers of support documents can also be observed in Figure 4.5, which illustrates the distribution of the number of support documents per sample in both datasets. WikiHop shows a Poisson-like behaviour, whereas MedHop exhibits a bimodal distribution, in line with our observation that certain drugs and

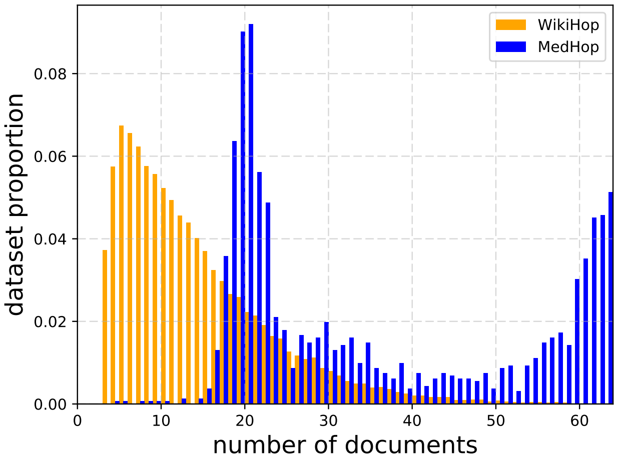


Figure 4.5: Support documents per training sample in both WIKIHOP and MEDHOP.

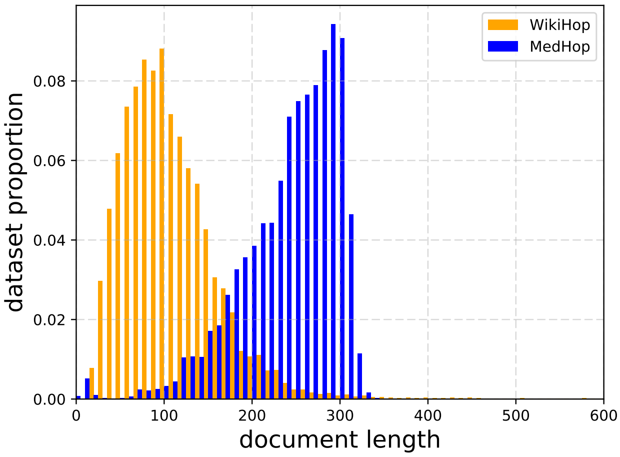


Figure 4.6: Histogram for document lengths in WIKIHOP and MEDHOP. Note that there is a long, but thin tail for WIKIHOP.

proteins have far more interactions and research papers associated with them. When we consider the distribution of document lengths in the datasets (see Figure 4.6), we again observe a Poisson-like distribution for WIKIHOP – note that these documents correspond directly to individual WIKIPEDIA article paragraphs. On the other hand, documents in MEDHOP are generally longer. The distribution clearly reflects the maximum length of 300 words (plus title) for research paper abstracts, as e.g. for the PLoS ONE journal, and a small number of documents with only a title but no abstract.

Query Type	**Proportion in Dataset**
instance_of	10.71 %
located_in_the_administrative_territorial_entity	9.50 %
occupation	7.28 %
place_of_birth	5.75 %
record_label	5.27 %
genre	5.03 %
country_of_citizenship	3.45 %
parent_taxon	3.16 %
place_of_death	2.46 %
inception	2.20 %
date_of_birth	1.84 %
country	1.70 %
headquarters_location	1.52 %
part_of	1.43 %
subclass_of	1.40 %
sport	1.36 %
member_of_political_party	1.29 %
publisher	1.16 %
publication_date	1.06 %
country_of_origin	0.92 %
languages_spoken_or_written	0.92 %
date_of_death	0.90 %
original_language_of_work	0.85 %
followed_by	0.82 %
position_held	0.79 %
Top 25	72.77 %
Top 50	86.42 %
Top 100	96.62 %
Top 200	99.71 %

Table 4.4: The 25 most frequent query types in WIKIHOP alongside their proportion in the training set.

Types of Queries Table 4.4 gives an overview over the 25 most frequent query types in WIKIHOP, alongside their relative proportion in the dataset. Overall, the distribution across query types has a long tail with rare types of query relations. The total number of query types in WIKIHOP is 277, MEDHOP on the other hand has only a single one: `interacts_with`.

Unique multi-step answer.	36%
Likely multi-step unique answer.	9%
Multiple plausible answers.	15%
Ambiguity due to hypernymy.	11%
Only single document required.	9%
Answer does not follow.	12%
WikiData/Wikipedia discrepancy.	8%

Table 4.5: Qualitative analysis of WikiHop samples.

4.5 Qualitative Analysis

After establishing these basic quantitative statistics of the two datasets, we next perform several qualitative analyses. Our aim is to examine the quality of the resulting data, and in particular revisit some of the assumptions made in the dataset induction method. We thus sample and manually annotate 100 examples from the development set of each of the two datasets.

4.5.1 WikiHop

A number of examples of the WikiHop dataset can be found in Table 4.6, which excludes document chains leading to distractor candidates for brevity. In Table 4.5 we list several qualitative attributes, and the corresponding proportion of samples in WikiHop.

The answer is not always uniquely determinable from the given text, in some cases it is merely suggested as the most likely. For example, the first sample from Table 4.6 suggests the correct answer by analogy. Among the 100 analysed samples, for a total of 45% the correct answer either follows as the unique answer from multiple texts directly, or is suggested as likely considering several texts. Furthermore, for a total of 26% of samples more than one candidate is plausibly supported as the correct answer from the given documents – including the correct answer. We observe that hypernymy is a frequent reason for this: the appropriate granularity level for the correct answer is not always clearly specified. For example, the query `(west suffolk, located_in_the_administrative_territorial_entity, ?)` could either have the candidate `suffolk` or `england` as the correct answer. These ambiguous samples show that the closed world assumption on the KB is only an ideal:

Query: (the big broadcast of 1937, genre, ?) **Answer:** musical film
Text 1: The Big Broadcast of 1937 is a 1936 Paramount Pictures production directed by Mitchell Leisen, and is the third in the series of Big Broadcast movies. The musical comedy stars Jack Benny, George Burns, Gracie Allen, Bob Burns, Martha Raye, Shirley Ross [...]
Text 2: Shirley Ross (January 7, 1913 – March 9, 1975) was an American actress and singer, notable for her duet with Bob Hope, "Thanks for the Memory" from "The Big Broadcast of 1938"[...]
Text 3: The Big Broadcast of 1938 is a Paramount Pictures musical film featuring W.C. Fields and Bob Hope. Directed by Mitchell Leisen, the film is the last in a series of "Big Broadcast" movies[...]

Query: (cmos, subclass_of, ?) **Answer:** semiconductor device
Text 1: Complementary metal-oxide-semiconductor (CMOS) [...] is a technology for constructing integrated circuits. [...] CMOS uses complementary and symmetrical pairs of p-type and n-type metal oxide semiconductor field effect transistors (MOSFETs) for logic functions. [...]
Text 2: A transistor is a semiconductor device used to amplify or switch electronic signals[...]

Query: (raik dittrich, sport, ?) **Answer:** biathlon
Text 1: Raik Dittrich (born October 12, 1968 in Sebnitz) is a retired East German biathlete who won two World Championships medals. He represented the sports club SG Dynamo Zinnwald [...]
Text 2: SG Dynamo Zinnwald is a sector of SV Dynamo located in Altenberg, Saxony[...] The main sports covered by the club are biathlon, bobsleigh, luge, mountain biking, and Skeleton (sport) [...]

Query: (minnesota gubernatorial election, office_contested, ?) **Answer:** governor
Text 1: The 1936 Minnesota gubernatorial election took place on November 3, 1936. Farmer-Labor Party candidate Elmer Austin Benson defeated Republican Party of Minnesota challenger Martin A. Nelson.
Text 2: Elmer Austin Benson [...] served as the 24th governor of Minnesota, defeating Republican Martin Nelson in a landslide victory in Minnesota's 1936 gubernatorial election.[...]

Query: (ieee transactions on information theory, publisher, ?)
Answer: institute of electrical and electronics engineers
Text 1: IEEE Transactions on Information Theory is a monthly peer-reviewed scientific journal published by the IEEE Information Theory Society [...] the journal allows the posting of preprints [...]
Text 2: The IEEE Information Theory Society (ITS or ITSoc), formerly the IEEE Information Theory Group, is a professional society of the Institute of Electrical and Electronics Engineers (IEEE) [...]

Query: (country_of_citizenship, louis-philippe fiset, ?) **Answer:** canada
Text1: Louis-Philippe Fiset [...] was a local physician and politician in the Mauricie area [...]
Text2: Mauricie is a traditional and current administrative region of Quebec. La Mauricie National Park is contained within the region, making it a prime tourist location. [...]
Text3: La Mauricie National Park is located near Shawinigan in the Laurentian mountains, in the Mauricie region of Quebec, Canada [...]

Table 4.6: Examples of relevant document combinations in WikiHop, connecting the entity in the query with the correct answer. The correct answers are underlined.

including type-consistent false answer candidates from Wikidata leads to some questions containing several true answer options, even though no corresponding facts are listed in Wikidata.

In 9% of samples we observe that a single document is already sufficient to identify the correct answer, without further background knowledge. These samples contain a document that states enough information about the query `item` and the answer together. In one such example, the query is `(Louis Auguste, father, ?)` with the correct answer `Louis XIV of France`, and a slight rewording `French king Louis XIV` is already mentioned in the document about `Louis Auguste`. In this case only relatively shallow paraphrasing is required, which is a consequence of imperfect entity linking, or – from a different perspective – of framing the task as an *extractive* RC task, where the answer has to be mentioned verbatim.

The task we pose is considerably more challenging than many prior tasks relying on distant supervision, yet we observe only 20% of samples in violation of the distant supervision assumption – a comparable fraction to other work (Riedel et al., 2010). Such violations can be the result of conflicts between WIKIDATA and WIKIPEDIA (8%), for example if different birth dates are recorded in WIKIDATA and the WIKIPEDIA article, or if the correct answer cannot be identified using the given documents (12%).

When trying to answer 100 of the questions ourselves with unlimited time per question, but no access to information other than the given documents, we achieved an overall accuracy of 74%. This human performance estimate may be influenced by concrete prior knowledge, although only to a limited degree: for 9% of samples the answer was already known even without the given documents. In addition, we further tested human accuracy on a validated portion of the development set (see Section 4.5.3). When answering 100 questions on this dataset, we achieved a human-level accuracy of 85%.

4.5.1.1 Crowdsourced Annotation

Besides our own qualitative analysis, we presented *Amazon Mechanical Turk* annotators with samples from the WIKIHOP development set. Annotators were shown the query-answer pair as a fact, and the chain of relevant documents leading to the answer. They were then instructed to answer i) whether they knew the fact before; ii) whether the fact follows from the given texts (with options *"fact follows"*, *"fact is likely"*, and *"fact does not follow"*); and iii) whether a single or several of the documents are required to infer the fact. Each sample was shown to three different annotators.

The annotators were familiar with the fact 4.6% of the time; we thus rule out prior knowledge of the given fact as a major factor which could affect the other judgments. The inter-annotator agreement, measured with Fleiss' kappa, is 0.253 in ii), and 0.281 in iii). This corresponds to a *fair* overall agreement, following the terminology established in prior work (Landis and Koch, 1977). Still, as there is non-negligible disagreement between annotators, a majority vote was used to aggregate annotations of the same data sample.

9.5% of the examples did not have a clear outcome of the majority vote in (2). Among those cases *with* a majority judgment, 59.8% are examples where the given fact *"follows"*, for 14.2% the fact is determined to be *"likely"*, and to *"not follow"* in 25.9%. These results are similar to the findings of our own annotation, and support the use of the distant supervision strategy, although it also shows that the dataset contains a significant portion of noisy samples.

Further analysing these annotations, among examples with a clear outcome of the majority vote in ii) of *"follows"* or *"likely"*, 55.9% of examples were judged by a majority as requiring several documents to infer the fact, and 44.1% stated that only a single document is required. The latter fraction is substantially larger than expected, given the dataset construction via graph traversal across several documents, including the direct filtering out of samples where the graph search starting point document already mentions the correct answer. But when further analysing cases judged as *"single"* in more detail, we found that they often indicate the correct answer in one of the documents, albeit without mentioning it literally. One such example is the fact `(witold cichy, country_of_citizenship, poland)` which has support documents d_1*: Witold Cichy (born March 15, 1986 in Wodzisław Śląski) is a Polish footballer[...]* and d_2*: Wodzisław Śląski [...] is a town in Silesian Voivodeship, southern Poland[...].* Here the information given in d_1 suffices for a human with the background knowledge that the attribute *Polish* is related to *Poland*, which obviates the need for further information in d_2 to establish the correct answer.

The previous example illustrates that whether or not a sample requires multi-hop inference is indeed a function of the background knowledge of the reader, as well as the set of implicit connotations they operate with. If the necessary information is already implicitly associated and accessible in the interpretation of the first document, no further document is needed to provide explicit information that helps infer the answer via a separate step. Thus, models with access to more connotations (e.g. via the distributional semantics learned in their pre-trained word-embeddings) may be able to solve a given sample without having to rely on other documents that provide explicit information, and instead leverage their own pre-trained associations.

4.5.2 MEDHOP

In terms of prior knowledge requirements MEDHOP is more complex than WIKIHOP, and there is a considerable number of documents in each sample (see Figure 4.5 in Section 4.4). We quickly observed that the workload of a human annotator to read *all* support documents for 100 samples is infeasible. Similar to the crowd-sourced annotation of WIKIHOP, we thus chose to analyse the dataset by considering only relevant documents – those visited on the path that arrives at the correct answer. The annotator analysed if the answer to the query *"follows"*, *"is likely"*, or *"does not follow"*, given the relevant documents. In total, 68% of the cases were considered as *"follows"* or as *"is likely"*, determined by an NLP researcher with prior experience in biomedical information extraction. One observation made during this analysis was that many cases which violate the distant supervision assumption are due to a missing PPI which is not stated in the connecting documents. This finding is encouraging, yet also shows the limitations of our dataset induction strategy, as it can result in a substantial number of noisy samples.

4.5.3 Validated Test Sets

Training models on data points with distant supervision can be successful, yet testing a model on noisy data is potentially problematic, as we would measure models partly by how well they fit noise. Ideally, the methods tested should thus also be evaluated on a manually validated test set. We hence extract parts of the test sets of WIKIHOP and MEDHOP for which human annotators judge that the answer is entailed by the given document. In this we differ from prior work which conducts evaluation only on distantly supervised samples (Hermann et al., 2015; Hill et al., 2016; Hewlett et al., 2016).

In the case of WIKIHOP, we follow the annotation method previously laid out in Section 4.5.1.1. We choose as validated test samples those which are labeled by at least 2 out of 3 annotators as *"follows"*, and also as requiring *"multiple"* documents. For MEDHOP on the other hand, crowdsourcing is not a feasible approach, as the domain demands a specialist background. As remarked before, both the number of texts given, as well as the length of each document are larger for MEDHOP than for WIKIHOP, which furthermore complicates annotation. Thus, only 20% of the MEDHOP test set was annotated, which is a small number of samples in absolute terms, yet can still give an indication for a model's accuracy on validated data, where the answer is implied by the text and where several documents are necessary.

After having establishing these human-validated test sets, we will next conduct a number of experiments on the WIKIHOP and MEDHOP datasets. We will compare established neural RC methods alongside several shallow statistical predictors, and analyse their behaviour in a variety of settings.

4.6 Experiments

We have constructed two datasets with a new dataset induction methodology intended to learn multi-hop reading comprehension behaviour across documents. In the previous section, we have discussed various properties of these datasets, analysed them, and established validated test sets for a better interpretation of experimental results.

How do neural RC models fare on these new datasets? What do they learn, and to what extent do they leverage information stemming from separate documents? We will next establish several model baselines for both WIKIHOP and MEDHOP, including two neural RC models alongside various shallow statistical predictors. We will begin by introducing the models we compare one by one, and then discuss their performance and behaviour in a variety of experimental settings. Models will be trained on the training set of either WIKIHOP or MEDHOP, whereas evaluation will be conducted both on the validated parts (Section 4.5.3), as well as the full test sets.

4.6.1 Models

Random Prediction This baseline predictor randomly chooses one of the given candidates. Recall that each sample generally has a different number of candidates.

Max-mention This model predicts the most frequently mentioned candidate in the given support documents S of a sample; ties are broken randomly. The *Max-mention* baseline allows us to observe whether the search procedure in the dataset assembly graph leads to imbalances on the level of candidate mentions, which could be a very simple signal for neural RC models to exploit.

Majority-candidate-per-query-type This baseline predicts the candidate $c \in C$ most frequently appearing as the correct answer on training samples, conditional on the query type of q. For WIKIHOP, this corresponds to the property of the given query, e.g `country`. For MEDHOP on the other hand, there exists by design only one type of query: `interacts_with`.

TF-IDF We have in the experiments on SciQ (in the previous chapter) already observed that Information Retrieval (IR) models can be strong baselines in QA tasks, an observation similarly made in prior work (Clark et al., 2016). IR baselines identify relevant documents based on lexical overlap with the question or query, though typically do not combine information from several documents besides prediction aggregation. We include a TF-IDF baseline in order to observe whether the correct answer can be identified from individual documents by exploiting lexical overlap of documents with the query and candidates. Concretely, this model forms its prediction as follows: each candidate is (separately) appended to the query, forming expressions $[q;c]$, which are given to the *whoosh* text retrieval system as *OR* query.[3] The system uses an inverted index of S, and returns TF-IDF similarity scores for each of these given support documents. The model then selects the candidate which scores highest in terms of TF-IDF value, across all support documents:

$$\arg\max_{c\in C}\,[\max_{s\in S}\,(\mathit{TF\text{-}IDF}([q;c],s))] \tag{4.1}$$

Document-cue We have previously described that particular combinations of documents and answers frequently co-appear in the training data. This predictive pattern is so prevalent that the correct answer can be inferred merely by the presence of a particular document among S. The *document-cue* baseline uses the predictor introduced in Section 4.2.2, measuring to what extent a model can exploit *cooccurrence* cues. As a reminder, this baseline predicts the candidate with highest *cooccurrence* score across the given support documents S:

$$\arg\max_{c\in C}\,[\max_{d\in S}(\mathit{cooccurrence}(d,c))] \tag{4.2}$$

All above models are shallow statistical predictors; they will serve as a useful reference point to gauge the performance of more sophisticated neural RC approaches.

[3] `https://pypi.python.org/pypi/Whoosh/`

Extractive neural RC models: *FastQA* and *BiDAF* The Bidirectional Attention Flow model (*BiDAF*, (Seo et al., 2017a)) and *FastQA* (Weissenborn et al., 2017) are extractive QA models based on an LSTM architecture. Both have demonstrated robust test set generalisation on SQUAD, and they form their prediction as a text span in the given document. It is worth pointing out that both models were developed and evaluated on single-hop RC datasets. They encode the document using *bidirectional* LSTMs, coupled with attention over the full text. This provides them – at least theoretically – with the capacity to condition the processing of textual information on information stated elsewhere, at a separate location in the document.

In order to adapt them to a setting with several documents, we concatenate all the given support documents $d \in S$, interleaved by separator tokens. As training target, the first mention of the correct answer in the concatenated super-document is used as the correct span, which the training loss is derived from. In a small side experiment we briefly evaluated randomly choosing the gold span among all mentions of the correct answer where there are several, but without observing significant differences. For evaluation we measure the models' prediction accuracy, which corresponds to the exact match (EM) score between the correct answer and the model prediction. Following the answer normalisation procedure in SQUAD (Rajpurkar et al., 2016) we lowercase both, remove articles, trailing whitespaces, and punctuation. We rule out the order of the concatenated documents as predictive cue by randomising it both during training and evaluation.

For the *BiDAF* model we follow the hyperparameter choice given in the implementation of the authors (Seo et al., 2017a) and use pre-trained *GloVe* (Pennington et al., 2014) word embeddings. We deviate from this setup in restricting the maximum length of the super-document to 8,192 words, use a hidden size of 20, and train the model for 5,000 iterations with batch size 16 to fit the full model into memory.[4] For *FastQA* we also use the implementation provided by the authors, again with pre-trained *GloVe* embeddings, not using character-embeddings, no maximum support length, hidden size 50, and batch size 64 for 50 epochs.

[4] Note that the concatenated super-document contains more tokens than the single WIKIPEDIA paragraph used in SQUAD, hence the demand for additional memory.

	WIKIHOP		MEDHOP	
Model	test	test*	test	test*
Random	11.5	12.2	13.9	20.4
Max-mention	10.6	15.9	9.5	16.3
Majority-candidate-per-query-type	38.8	44.2	**58.4**	**67.3**
TF-IDF	25.6	36.7	9.0	14.3
Document-cue	36.7	41.7	44.9	53.1
FastQA	25.7	27.2	23.1	24.5
BiDAF	**42.9**	**49.7**	47.8	61.2

Table 4.7: Test accuracy in [%] across models for the WIKIHOP and MEDHOP datasets. Columns marked with asterisk are for the validated portion of the dataset.

4.6.2 Model Comparison

Table 4.7 summarises the experimental outcomes for WIKIHOP and MEDHOP, on both the full as well as the validated portion of the respective test sets. We first observe that the predictions of the *max-mention* baseline are similarly accurate as those of the *random* baseline. That is, the candidate mention frequency is not a predictive cue to identify the correct answer.

Forming answer predictions based on the frequency with which a candidate was observed in the training set, however, reaches 38.8% / 44.2% and 58.4% / 67.3% accuracy on WIKIHOP and MEDHOP, for the full / annotated test samples, respectively. This means that a comparatively simple statistic, which only exploits the frequency of particular answers to query types on the training set, forms a relatively strong predictor; it even reaches the highest overall accuracy on MEDHOP. Note that the accuracy of 58.4% for MEDHOP, where only a single query type is present, does not signify that 58.4% of the test set have the same answer. Instead, we emphasise that this baseline is restricted to the *given* candidates C in a sample. Among these, the relative frequency with which different candidates have been observed as the correct answer for training samples is a strong predictor for the correct answer at test time.

The information retrieval TF-IDF model outperforms the random baseline on WIKIHOP, but is overall less predictive than some of the other shallow predictors. That is, the tokens in the query are useful to identify documents mentioning the answer, even though documents mentioning *both* query subject *and* answer in the same document are excluded from the dataset. This highlights a limitation of our mention-centric approach to interpreting multi-hop behaviour in dataset assembly: where entities are not explicitly co-mentioned in the same document, it is assumed that a second document is necessary to answer the query. The TF-IDF baseline, however, shows that even with a single document, lexical overlap can provide relevant cues to identify documents mentioning the correct answer, at least to a moderate extent. On the other hand, as drug names in MEDHOP are normalised, no interacting drug pair is mentioned together in the same document, hence the TF-IDF baseline reaches even lower accuracy than random predictions. In summary, lexical overlap with an individual given document is a weak but informative cue on WIKIHOP, and an insufficient basis for accurate predictions on MEDHOP.

As the last shallow predictor baseline, the *document-cue* method predicts 36.7% / 41.7% and 44.9% / 53.1% of WIKIHOP and MEDHOP samples correctly, on the two respective test sets for each. Note that this is despite sub-sampling the data according to the frequency of pairs of particular documents and answers in WIKIHOP. In MEDHOP, where this sub-sampling was not conducted due to the much smaller dataset size, this predictor can correctly solve more than half of the samples on the annotated test set portion.

For the two neural RC models, *BiDAF* achieves higher accuracy values than *FastQA* on both WIKIHOP and MEDHOP. This is unexpected considering the similarity of their respective performance on the SQUAD dataset. Possible explanations for this are the better ability of *BiDAF* to process rare words due to its character-level representations, as well as the use of several layers of latent interaction in the *BiDAF* architecture. Compared to SQUAD this may be of increased importance in our task, where relevant information is spread across various locations in the super-document. Note that both models select the answer by predicting a span in one of the given documents, and without direct access to the set of given candidate options C.

Finally, we generally observe that the results on the validated test set por-

Model	**Unfiltered**	**Filtered**	**Δ**
Majority-candidate-per-query-type	41.2	38.8	-2.4
TF-IDF	43.8	25.6	-18.2
Document-cue	74.6	36.7	-37.9
Train set size	527,773	43,738	-91.7

Table 4.8: Accuracy comparison (in [%]) for three shallow statistical predictor models on WIKIHOP, *before* and *after* filtering the data. The value of Δ corresponds to the absolute percentage difference (for the dataset sizes: the relative difference).

tions correlate strongly with those on the full test set. There is furthermore an improvement from the full (noisy) test accuracy to the accuracy observed on the validated test samples, consistently across both datasets and all models. This suggests that i) even the noisy test sets can give a good indication of relative model abilities ii) the respective training sets – even though noisy – contain a signal which is strong enough to generalise to validated held-out evaluation samples, and iii) it is more difficult to predict the correct answer for noisy samples.

4.6.3 The Effect of Sub-Sampling on Statistical Predictors

We have observed that the relative strength of shallow statistical predictors is an important problem that needs to be addressed when assembling multi-hop RC datasets, and we have used sub-sampling to mitigate the problem (cf. Section 4.2.2). To quantify the effectiveness of this, we next compare several statistical predictors on WIKIHOP, both before and after filtering; results can be found in Table 4.8. The strong performance of the different predictors, especially of the *document-cue* baseline before filtering demonstrates that addressing dataset biases is critical: otherwise 74.6% accuracy could be reached by only exploiting $cooccurrence(d,c)$. This also emphasises the significance of examining and circumventing potential dataset biases, especially when engaging in automatic RC dataset assembly procedures, where natural structural regularities can result in shortcuts to solving the task.

While we did not fully rid the datasets of these shallow predictive cues, it is important to be aware of these as they could otherwise lead to misleading interpretations based on model accuracy. The performance drop after filtering shows that sub-sampling can successfully reduce the extent of the issue, albeit with a substantial reduction in terms of dataset size. Hence the application of filtering on smaller datasets – like MEDHOP– is problematic; we will however see that a different method can provide remedy, which we will next investigate.

4.6.4 Candidate Masking

There exist other methods beyond sub-sampling to remove shallow cues from a given RC dataset, such as randomly masking answer expressions. Lexical cues among possible answers can be a problem, as previously described by Hermann et al. (2015). It is worth noting that all of the shallow predictors introduced here rely on the explicit identity of the given answer candidates, in order to link them to observations made on the training set. Arguably though, in RC the correct answer to a comprehension query should be formed from the textual *context* that the answer is surrounded with, rather than from an intrinsic property or connotation of the answer text itself that identifies it as the answer to the query.

As we intend to measure a model's ability to form a prediction based on the textual context surrounding the given candidate mentions, we next conduct experiments in which we transform the dataset by *masking* answer candidates. Concretely, we substitute any candidate mention randomly with one of 100 placeholder tokens – for example *"Mumbai is the most populous city in* MASK7." We mask candidate mentions consistently within a sample to preserve coreference, but the placeholder tokens generally differ if the same candidate appears in another sample. The same set of 100 masking tokens is used in both training and evaluation sets to avoid potential new out-of-vocabulary effects at test time. Masking effectively avoids cues stemming from the relative frequency of a candidate, as well as correlations of particular documents and candidates. Models are thus unable to form predictions based on properties of the answer itself, and instead have to form them by considering the surrounding text in which the answer candidates are embedded.

Results for experiments in this masked data version of the task can be found in Table 4.9. We observe that the baseline predictors, which rely on lexical information, deteriorate drastically compared to Table 4.7. This is encouraging in particular for MEDHOP, for which sub-sampling does not constitute a viable option due to

	WIKIHOP		MEDHOP	
Model	test	test*	test	test*
Random	12.2	13.0	14.1	22.4
Max-mention	13.9	20.1	9.2	16.3
Majority-candidate-per-query-type	12.0	13.7	10.4	6.1
TF-IDF	14.4	24.2	8.8	14.3
Document-cue	7.4	20.3	15.2	16.3
FastQA	35.8	38.0	31.3	30.6
BiDAF	**54.5**	**59.8**	**33.7**	**42.9**

Table 4.9: Test accuracies for the WIKIHOP and MEDHOP datasets in the masked data setup. Columns marked with asterisk are for the validated portion of the dataset.

its already relatively small size. Masking answer candidates thus forms an effective approach to circumvent statistical cues which neural networks could otherwise learn to exploit.

An interesting observation is that the two neural models can maintain or even increase their overall accuracy in the masked setting. Both models, and in particular *BiDAF* now substantially outperform all shallow heuristics. That is, the neural RC models successfully exploit the text surrounding the candidate mentions to infer the correct answer.

It is worth noting that the drug entities mentioned in MEDHOP are normalised, i.e. represented by a particular identifier unique to a given drug. Randomising this information, as is done in the masked setting, thus results in overall decreased performance compared to the unmasked setting. On the other hand, the answer candidates in WIKIHOP frequently resemble multi-word expressions. Together with the wider range of query types, the reduced vocabulary size of only 100 single-word placeholder tokens in the masked setting thus simplifies the span prediction task on WIKIHOP.

Finally, while both neural models achieve higher accuracy than the statistical predictors, both leave a substantial gap to human accuracy (74% / 85% for WIKIHOP, cf. Section 4.5.1). We will next conduct two ablation studies which will help us further understand the behaviour of the two neural RC models, as well as suggest a promising avenue for improving model performance.

	WikiHop				MedHop			
	standard		**masked**		**standard**		**masked**	
Model	test	test*	test	test*	test	test*	test	test*
BiDAF	42.9	49.7	54.5	59.8	47.8	61.2	33.7	42.9
BiDAF oracle	57.9	63.4	81.2	85.7	86.4	89.8	99.3	100.0
BiDAF : Δ	+15.0	+13.7	+26.7	+25.9	+38.6	+28.6	+65.6	+57.1
FastQA	25.7	27.2	35.8	38.0	23.1	24.5	31.3	30.6
FastQA oracle	44.5	53.5	65.3	70.0	54.6	59.2	51.8	55.1
FastQA : Δ	+18.8	+26.3	+29.5	+32.0	+31.5	+24.7	+20.5	+24.5

Table 4.10: Comparison of test accuracy in [%] when giving models oracle access to the documents directly leading to the correct answer. Δ describes the difference, again in [%], between standard and oracle setting. Columns with asterisk again hold results for the validated samples.

4.6.5 Oracle Access to Relevant Documents

The samples in both WikiHop and MedHop contain not only documents along paths leading to correct answers, but also irrelevant documents along paths that lead to false answer candidates. We will next investigate the performance of *BiDAF* and *FastQA* in an oracle setting where models are given only a subset of the documents in S. Concretely, the models are given those documents which were traversed along the way to the correct answer in the graph traversal during dataset induction. We can thus examine the accuracy possible if the RC models were capable of selecting these relevant documents, both during training and evaluation.

The results of this experiment can be found in Table 4.10. We observe that both models, across all settings, improve considerably when given document oracle information, achieving up to 81.2% / 85.7% accuracy on WikiHop, and 99.3% / 100.0% on MedHop (masked setting; *BiDAF*). This shows that the models are able to predict the correct answer with reasonable accuracy in a scenario where fewer or no alternative distractor candidates are mentioned. In particular for MedHop, where the given abstracts often center around only one particular drug candidate, the model can then achieve 100% accuracy. In the masked setting, the RC models are likely able to select masks as candidates and reach nearly 100% accuracy, though interestingly only *BiDAF* is able to learn this. The results furthermore underscore the importance of introducing negative candidates and relevant distractor documents, since type consistency alone is a very important cue for the task. When using false candidates and relevant distractor documents, models have to consider alternative answer possibilities and their respective surrounding document context. Thus, when inverting the interpretation direction, the robust improvements we see in the oracle setting can conversely be interpreted as a deterioration: across all setups we find that the models' answer prediction process can be fooled by the inclusion of documents leading to other, type-consistent alternative answer candidates.

Finally, these results also indicate that the selection of relevant texts may be a promising directions for further model development, a route we will embark on in the subsequent chapter.

4.6.6 Ablation: Removing Relevant Documents

We next conduct an ablation study to investigate if *BiDAF* and *FastQA* utilise information from several documents when forming their prediction. To this end we conduct an experiment in which we remove the first document traversed in the graph search, which mentions the entity in the query, and furthermore all documents not mentioning any answer candidate. This way we i) remove information one would expect to be relevant when performing multi-hop inference ii) retain the document mentioning the correct answer, which is necessary for the extractive RC models to be able to answer correctly, and iii) avoid introducing imbalances between candidates, by treating documents leading to distractors in the same way as documents on paths leading to the correct answer.

Table 4.11 lists the results of this ablation. We observe a moderate and consistent drop in model performance for the *BiDAF* model: the difference is 10.0%/2.1% on WIKIHOP and 3.3%/6.2% on MEDHOP. This suggests that *BiDAF* is to a small extent leveraging information from multiple documents when it forms its prediction. *FastQA* on the other hand behaves inconsistently; it shows a slight increase of 2.2%/3.2% for WIKIHOP, and a slight decrease of 2.7%/6.1% on MEDHOP, and results are thus inconclusive overall.

This concludes our experiments for the WIKIHOP and MEDHOP datasets. We

	WIKIHOP		MEDHOP	
Setup	test	test*	test	test*
BiDAF	54.5	59.8	33.7	42.9
BiDAF (doc's removed)	44.6	57.7	30.4	36.7
FastQA	35.8	38.0	31.3	30.6
FastQA (doc's removed)	38.0	41.2	28.6	24.5

Table 4.11: Test accuracy (masked), both in the standard setting, as well as when only documents containing answer candidates are given (doc's removed).

have found that shallow statistical predictors can be strong baselines compared to more sophisticated neural RC methods. However after anonymising answer candidates, neural RC models are able to retain their performance, which the simple heuristics could not. Neural RC models benefit from direct oracle access to relevant document sets, and the prediction mechanism in *BiDAF* partly has learned to rely on several documents when forming its prediction, although only to a moderate extent. We will continue and conclude this chapter with a broader discussion of prior work on multi-hop inference in NLP, and how our work on WIKIHOP and MEDHOP relates to it.

4.7 Discussion: Prior Work on Multi-Hop Inference

Related Datasets WIKIHOP and MEDHOP can be conceived as text-based QA datasets. In this regard they are related to prior datasets, for example based on FREEBASE (Berant et al., 2013; Bordes et al., 2015), WIKIPEDIA (Yang et al., 2015; Rajpurkar et al., 2016; Hewlett et al., 2016), web search queries (Nguyen et al., 2016), news articles (Hermann et al., 2015; Onishi et al., 2016), books (Hill et al., 2016; Paperno et al., 2016) or trivia (Boyd-Graber et al., 2012; Dunn et al., 2017). Apart from TRIVIAQA (Joshi et al., 2017), all the above datasets use single documents, and text comprehension typically does not involve a combination of several independent facts. WIKIHOP and MEDHOP are, in contrast, constructed to target multi-hop inference and cross-document RC.

Prior to MEDHOP and WIKIHOP other multi-hop RC resources have been assembled, but these have a very limited number of samples (e.g. the FRACAS test suite), or are constructed using synthetic language (Weston et al., 2016). While TRIVIAQA samples frequently involve multi-step inference, the complexity mostly lies in the interpretation of compositional questions. In contrast, the WIKIHOP and MEDHOP datasets target inference from several documents, yet with relatively simple queries. Furthermore, through the use of multiple documents, the multi-hop inference required extends beyond coreference resolution.

The subsequently developed COMPLEXWEBQUESTIONS (Talmor and Berant, 2018) and HOTPOTQA (Yang et al., 2018) datasets also address multi-hop RC, and contain questions which are posed in natural language form. A core aspect of these two datasets is the need for compositional and conjunctive query interpretation (e.g. "*What city is the birthplace of the author of 'Without end', and hosted Euro 2012?*" for the former, and "*What was the former band of the member of Mother Love Bone who died just before the release of "Apple"?*" for the latter). The more involved query interpretation approach of these datasets stands in contrast to the datasets induced with our strategy, which instead emphasises the inference of answers from distributed textual evidence to structurally simple queries.

Compositional Knowledge Base Inference Rule-based inference with several facts is more commonly found in the realm of symbolic reasoning with structured information. These methods include Inductive Logic Programming (Quinlan, 1990; Pazzani et al., 1991; Richards and Mooney, 1991) and Markov Logic (Richardson and Domingos, 2006; Schoenmackers et al., 2008) – which can struggle with inefficient inference and limited coverage, although efforts to reduce sparsity have been undertaken, e.g. based on web text (Schoenmackers et al., 2010). A scalable method for learning compositional rules is the Path Ranking Algorithm (PRA) (Lao and Cohen, 2010; Lao et al., 2011), which leverages random walks in the entity graph, thus identifying salient graph paths. Gardner et al. (2013) circumvent the problem of graph sparsity through the introduction of additional virtual links based on dense vector similarity. Beyond that, a number of methods with other vector composition functions have been considered, for example vector addition (Bordes et al., 2014), RNNs (Neelakantan et al., 2015; Das et al., 2017), and memory networks (Jain, 2016). Another approach is the Neural Theorem Prover (Rocktäschel and Riedel, 2017), which uses dense rule and symbol embeddings to learn a differentiable backward chaining algorithm.

The above approaches focus on the problem of learning to combine KB facts of a particular structure and predefined schema. They can either be applied on the outputs of an IE system (Banko et al., 2007) or on human-annotated facts (Bollacker et al., 2008). The former can be noisy whereas the latter costly to acquire; both are typically incomplete and potentially biased in coverage.

The progress on neural RC models from the previous years has however shown that end-to-end models for natural language comprehension are capable of identifying answers to freely formulated comprehension questions from unstructured text directly, thus circumventing potential intermediate stages of question parsing or information extraction. With the datasets we have created, our aim is to help understand if neural RC models are capable of learning to process unstructured documents directly, while at the same time performing the type of inference and combination of information usually found in the context of logical inference on structured facts.

Text-Based Multi-Step Reading Comprehension Prior work by Fried et al. (2015) has shown that leveraging information from other documents can be beneficial for re-ranking predictions in an open-domain QA task. Another approach, in the context of science exam questions, is to form chains of textual background knowledge, which has the additional advantage of providing concrete explanations for the answer (Jansen et al., 2017). A variety of neural network models, tailored towards multi-hop RC, has been developed beyond that. This includes memory networks (Weston et al., 2015; Sukhbaatar et al., 2015; Kumar et al., 2016), which attend over memory items multiple times, and which have shown encouraging performance on synthetic multi-hop reasoning tasks (Weston et al., 2016). A common attribute of these neural approaches to multi-hop inference is their conditioning structure which enables the interaction and matching of representations for the question, the context, potential answer candidates, as well as combinations of these (Peng et al., 2015; Weissenborn et al., 2017; Xiong et al., 2017; Liu and Perez, 2017), which can be repeated in multiple iterations (Sordoni et al., 2016; Neumann et al., 2016; Seo et al., 2017b; Hu et al., 2017) and which can include learning to halt (Graves, 2016; Shen et al., 2017).

Learning Search Expansion Another related research direction considers the expansion of document sets available in a QA task, either using web navigation (Nogueira and Cho, 2016), query reformulation, or reinforcement learning (Narasimhan et al., 2016; Nogueira and Cho, 2017; Buck et al., 2018). This conceptually related research aims at adapting queries to improve the availability of relevant documents, similar to the expansion of support documents used for our datasets.

4.8 Conclusion

In this chapter we have defined a cross-document multi-hop Reading Comprehension task. We have described a general methodology for dataset induction, which we have used to construct datasets in two domains, and conducted a series of experiments to test several baseline methods. We now summarise the answers to the research questions posed in the beginning of this chapter:

1. **How can we construct datasets for multi-hop Reading Comprehension across documents?** We have developed a methodology for creating such datasets which requires both a structured knowledge base and a text corpus, ideally aligned in their domain. The method relies on a bipartite graph of mention-relationships between entities and documents to assemble chains of documents that link entity pairs across several document contexts. The method is automatic and relies on distant supervision, and we additionally relied on human annotations to validate parts of the resulting evaluation sets.

2. **What are potential dataset biases and pitfalls associated with the dataset assembly approach chosen?** There are several potential problems, which we could partly address, though not fully. The problems we have discussed include: i) the exploitation of type-consistency heuristics, which we have ad-

dressed through the inclusion of documents with alternative candidates; ii) label imbalance, which we have addressed through sub-sampling and randomised answer masking; iii) spurious correlations between particular documents and answers, which we have also addressed through sub-sampling and answer masking; iv) failure cases of the distant supervision assumption, which we have (at least during evaluation) mitigated through additional dataset validation.

3. **How do two neural RC models – *FastQA* and *BiDAF* – perform on the cross-document multi-hop RC datasets, in particular compared to shallow statistical baselines?** The two models achieve lower accuracy than on SQUAD, whereas *BiDAF* generally scores above *FastQA* on both WIKIHOP and MEDHOP. The performance of both does not exceed the accuracy of shallow statistical baselines by much (if at all), but in contrast to these the two neural models are able to maintain their performance when answer candidates are randomly masked. Both models improve their accuracy when removing distracting documents from the given support, and the accuracy of *BiDAF* shows a modest deterioration when removing some of the relevant documents, indicating the use of information from multiple documents.

The dataset assembly method in this chapter suffers from several limitations, which we have addressed and partially mitigated, but also has the strengths of cost-effectiveness and scalability. The datasets contain structured queries about entities, and they assume that the correct answer is mentioned literally in one of the documents. On the one hand this constrains the scope of possible queries and answers, but it also facilitates the assembly of the dataset, and furthermore simplifies evaluation. We see our study as a step towards models which learn to integrate cross-document information, and believe the construction of relevant datasets for this purpose to be an important intermediate objective which is worth further pursuit.

Subsequent to the assembly of the WIKIHOP and MEDHOP datasets, the NLP field has advanced considerably. A new generation of neural models, which leverage pre-trained representations fitted on large text corpora (Peters et al., 2018; Devlin et al., 2019; Liu et al., 2019), has led to substantial performance improvements across tasks – and notably also on RC datasets. It is worth emphasising that pre-trained models can learn factual information from the corpus they are pre-trained on (Petroni et al., 2019), which raises the question to what extent they can apply such prior knowledge on a dataset like WIKIHOP (which is developed based on WIKIPEDIA – which is itself typically included in pre-training corpora).

As pointed out in Section 2.3, prior knowledge is a central aspect of comprehension, and the prior knowledge implicit in pre-trained representations can conceptually alter the multi-hop aspect of RC, i.e. whether predictions are formed based on several pieces of text. In a concrete example, relevant factual knowledge about entities in WIKIHOP, e.g. *Mumbai*, may be implicit in the representations of these entities: *Mumbai* shares distributional context with *India*, leading to predictive associations between these two entities. If such information is available in pre-trained embeddings, then they obviate the need for additional declarative facts which state such information explicitly, e.g. *"Mumbai [...] is the most populous city in India"*. Implicit prior knowledge can thus potentially replace explicitly stated information in the comprehension process, and thus blur the conceptual boundaries of multi-hop comprehension. To what extent this is indeed the case in pre-trained models is however currently an open question.

Chapter 5

Selecting Document Combinations for Reading Comprehension across Documents

5.1 Overview

In the previous chapter we have tested the baseline performance for a number of different models in the multi-hop RC setting. In one of the experiments (Section 4.6.5), we have identified the benefit of direct oracle access to relevant documents. But how can an RC model distinguish the relevant documents from the irrelevant and distracting ones and identify the content that it needs to infer the answer? The following study will revolve around this question, and we will investigate different methods for selecting relevant document combinations for an RC model to read.

Selecting relevant pieces of text that convey information necessary to answer a comprehension question is of central importance in the RC task – even more so when multiple documents are involved. Text selection is also a sub-problem in text-based QA systems, and is typically achieved using Information Retrieval (IR) to efficiently gather documents from a large inverted document index. It would be computationally very burdensome to apply a costly neural RC model on a large document collection without previously filtering it – a problem that is further exacerbated if the model representations are computed conditional on each new question. Thus, especially when scaling up the corpus of documents under consideration (Chen et al., 2017a), IR can become a critical practical necessity.

Empirically, IR has been shown to be a performance bottleneck in NLP pipeline systems that apply a natural language understanding (NLU) component on a large text corpus. For example, for the DAM (Parikh et al., 2016) natural language inference model in Fact Checking, (Thorne et al., 2018) report an absolute accuracy drop of 37.63% when relying on outputs of an upstream TF-IDF retriever, rather than the text selection of an oracle.[1] In Open-Domain QA, Chen et al. (2017a) found that QA accuracy more than halves (from 69.5% to 27.1% exact match) when the RC model component has to rely on the noisy outputs of an IR system, rather than the relevant texts directly.[2] Finally, Yang et al. (2018) report more than 10% F_1 improvement on HOTPOTQA when using an oracle to provide multi-hop gold evidence to a subsequent RC component. Such performance gaps underscore the critical role that selecting relevant text (parts) plays: as NLU models are typically employed downstream of an IR component, they hinge on the successful retrieval of relevant facts or documents, or otherwise suffer from cascading errors.

In our previously conducted multi-hop experiments from Chapter 4, we have observed an absolute accuracy improvement of 15.0% when relevant document sets are directly given (WIKIHOP, *BiDAF*). This highlights the role of adequate content selection as performance bottleneck also for this task, and suggests that a mechanism for selecting relevant documents might – similar to open-domain QA – be a promising avenue to improve model performance also on WIKIHOP.

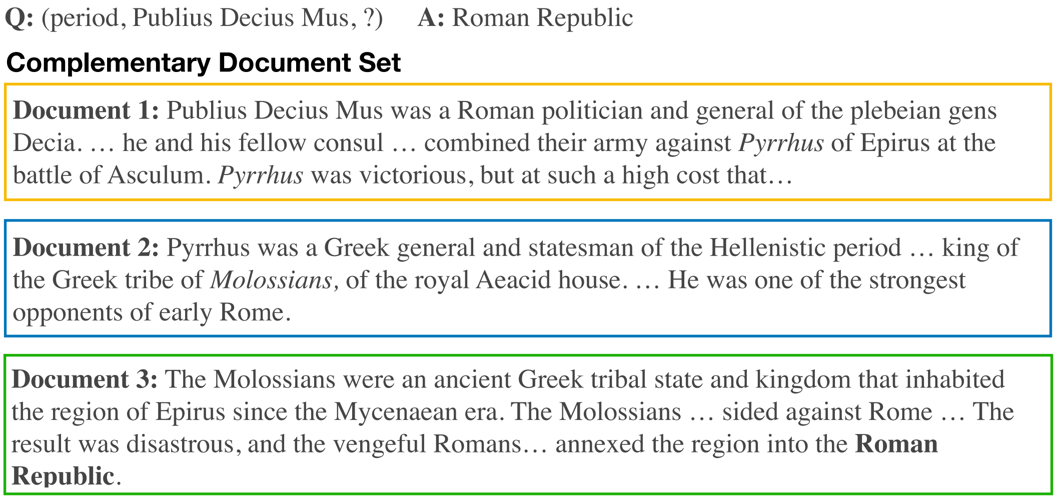
5.1.1 Limitations of TF-IDF as Selection Approach

Prior work in the RC context frequently uses TF-IDF vector similarity to determine document relevance (Dhingra et al., 2017; Clark and Gardner, 2018). The selection of relevant text in WIKIHOP is however very different, compared to e.g. SQUAD.

In the case of SQUAD, task-relevant information is locally very concentrated: prior work found that for 90% of samples in SQUAD a single sentence conveys

[1] Tables 3 and 4 in (Thorne et al., 2018); ‘RS’ setting.

[2] Numbers taken from Tables 4 and 6 in (Chen et al., 2017a).



Q: (period, Publius Decius Mus, ?) **A:** Roman Republic

Complementary Document Set

Document 1: Publius Decius Mus was a Roman politician and general of the plebeian gens Decia. … he and his fellow consul … combined their army against *Pyrrhus* of Epirus at the battle of Asculum. *Pyrrhus* was victorious, but at such a high cost that…

Document 2: Pyrrhus was a Greek general and statesman of the Hellenistic period … king of the Greek tribe of *Molossians,* of the royal Aeacid house. … He was one of the strongest opponents of early Rome.

Document 3: The Molossians were an ancient Greek tribal state and kingdom that inhabited the region of Epirus since the Mycenaean era. The Molossians … sided against Rome … The result was disastrous, and the vengeful Romans… annexed the region into the **Roman Republic**.

Figure 5.1: Problem illustration: selecting a relevant combination of documents for text comprehension. TF-IDF-based retrieval is not well-suited and only retrieves the first of the three documents in this example.

sufficient information to correctly answer the given question (Min et al., 2018). Naturally, identifying this sentence is of crucial importance to finding the correct answer. Furthermore there is substantial lexical overlap between question and sentence, which provides a strong clue for relevance: 81.2% Hits@1 can be achieved using a TF-IDF sentence selector alone (Min et al., 2018). Neural RC models for SQuAD thus benefit from adding explicit features that mark lexical overlap between question and text (Weissenborn et al., 2017; Chen et al., 2017a), even though they also possess the ability to *learn* to soft-match relevant text pieces using attention structures based on dense dot-product similarity, as e.g. in the *BiDAF* architecture (Seo et al., 2017a). In summary, lexical overlap to the question gives a direct cue for relevancy in SQuAD, thus rendering TF-IDF a potent method for selecting relevant content.

But the situation for WikiHop is different: here the lexical overlap between the given queries and documents mentioning the answer is reduced *per design*; thus TF-IDF alone is likely less useful. TF-IDF furthermore provides scores for the relevance of documents to the question that are computed independently of one another, given the query, thus providing individual sources of textual evidence whose relevance is considered in isolation. Figure 5.1 shows an example query where retrieving documents independently falls short at providing the necessary context for an RC model to read: the query about the period that *Publius Decius Mus* lived in can be used to retrieve *Document 1*, but fails to select *Document 3* which mentions the answer, as there is no lexical overlap with the query. This illustrates the limitations of TF-IDF when aiming to retrieve complementary documents for queries like those in WIKIHOP: TF-IDF considers document relevance in isolation and uses only lexical overlap with the question, but no further information from the other documents. TF-IDF thus imposes limitations on the types of questions that can then be answered (or facts that can be checked) by an NLU system – especially if the relevant information is not comprehensively expressed within a single document or sentence. Specifically in the multi-hop setting of WIKIHOP, with chains of documents inter-linked by entities, it would be desirable to develop a text selection mechanism for *combinations* of documents, where the relevance of document combinations is determined jointly by the constituent documents. In summary, the different structure of WIKIHOP samples suggests that TF-IDF might not be as effective a text selection method as elsewhere, hence it is worth considering alternatives.

5.1.2 Chapter Overview

We will in this chapter investigate the usefulness of different text selection mechanisms to address the problem of choosing relevant document combinations for RC on WIKIHOP. Motivated by the sample structure of WIKIHOP, we will investigate the use of pseudo-relevance feedback (PRF) as extension to TF-IDF, and show its empirical usefulness in selecting relevant documents. PRF can – broadly speaking – retrieve documents dependent on the content of other, previously retrieved documents, and is different from TF-IDF in this regard. We will see that PRF compares favourably to other retrieval methods, both in its ability to retrieve relevant documents, and also when coupled with the previously tested downstream RC models: *FastQA* and *BiDAF*. When further extending text selection with a trainable ranker for scoring document combinations, we can additionally improve the answer predictions, demonstrating the usefulness of selecting appropriate content for a downstream RC model to read.

But while PRF-based document combination ranking shows promising improvements on WIKIHOP, interestingly it does not result in improvements for HOTPOTQA (Yang et al., 2018), a more recently developed crowdsourced multi-hop dataset also based on WIKIPEDIA. We relate the differing model behaviours to different characteristics of the two datasets, highlighting that different dataset induction strategies result not only in different dataset biases (as previously observed and analysed for WIKIHOP in Chapter 4), but also affect the suitability of text selection components. We summarise the research questions addressed in this chapter with the following list.

List of Research Questions:

1. How does Pseudo-Relevance Feedback (PRF) compare to other IR methods when retrieving relevant documents on WIKIHOP?

2. How can PRF be adapted to retrieve document combinations, rather than individual documents?

3. How can document combinations retrieved with PRF be integrated with a trainable ranking model and a neural RC system, and how does this translate into downstream answer prediction performance on WIKIHOP?

4. How do observations on document selection for WIKIHOP compare with observations for HOTPOTQA; can they be related to different dataset induction choices?

5.2 Prior Related Work

Integrating Reading Comprehension with Information Retrieval on a large corpus of unstructured documents has been coined *Machine Reading at Scale* (MRS) (Chen et al., 2017a). The *DrQA* model for MRS (Chen et al., 2017a) presents a concrete open-domain QA system that uses TF-IDF to retrieve WIKIPEDIA documents as knowledge source, and combines them with a neural RC model. While *DrQA* has demonstrated the usefulness of neural RC approaches in a QA system on a larger corpus, it has also highlighted that the IR component is the main performance bottleneck for such a system. Even then, given the degree of lexical overlap in SQUAD questions with the relevant text (Min et al., 2018), TF-IDF-based search approaches may work better in an MRS setting than other, more general questions composed without prior exposure to the paragraph stating the answer – similar to our observations on SCIQ in Chapter 3.

One avenue towards improving the MRS pipeline is to tune the IR component further towards the objective of predicting the correct answer, hence increasing the relevance of the textual material read by the RC component. Such approaches can be based on joint training with reinforcement learning (Wang et al., 2018b), reranking (Wang et al., 2017; Htut et al., 2018), multi-task learning (Nishida et al., 2018), or improving the integration of answer predictions derived from several paragraphs (Clark and Gardner, 2018).

Expanding the set of retrieved documents can generally be achieved using Query Expansion (QE) (Azad and Deepak, 2017), for example by identifying and leveraging word relationships in a corpus, or by analysing documents retrieved by the initial query (Xu and Croft, 1996). Queries can be rewritten based on a knowledge base and textual entailment (Musa et al., 2018), reinforcement learning to issue new queries (Narasimhan et al., 2016), or by decomposing complex questions into a collection of simpler questions (Talmor and Berant, 2018). To navigate in a larger source of unstructured text, a model can structure documents as trees and train agents to navigate towards the relevant parts (Geva and Berant, 2018). Other work answers graph queries using a distribution over paths in a document graph (Chen et al., 2018), or translates QA into the search for an optimal sub-graph (Khashabi et al., 2018b).

5.3 Retrieving Combinations of Documents

A commonly used approach when applying a neural NLU component on multiple potentially relevant documents is to concatenate them into large super-documents (Thorne et al., 2018; Yang et al., 2018). This way, complementary pieces of text are still at least in principle available to the neural model component for joint processing. However, the approach has major drawbacks: concatenating documents does not scale; the total time and memory footprint for neural encoders becomes prohibitively large as more documents are considered. Moreover the recency bias of RNN-based RC models is not well-suited for long-range dependencies which arise in super-documents of complementary parts. For encoders which do consider long-range dependencies, such as transformers (Vaswani et al., 2017), scalability is quadratic, and thus also computationally problematic.

Decomposing the given textual evidence on the other hand – i.e. considering smaller groups of documents – and subsequently aggregating RC predictions, can overcome the aforementioned shortcomings. Binary relevancy judgements could efficiently be computed with a comparatively shallow model for many different smaller groups of documents, keeping only the most likely combinations as much shorter inputs for a more computationally expensive neural reader. A downside to evidence decomposition, however, is that the number of possible document combinations grows combinatorially in the number of documents used in a group, quickly approaching ranges that render the approach computationally impractical as well. In order to reduce this otherwise infeasible number of document combinations under consideration, an efficient selection mechanism for *combinations* of documents has to be applied. We will develop such a mechanism based on Pseudo-Relevance Feedback, and later couple it with a trainable document combination ranker.

5.3.1 Pseudo-Relevance Feedback

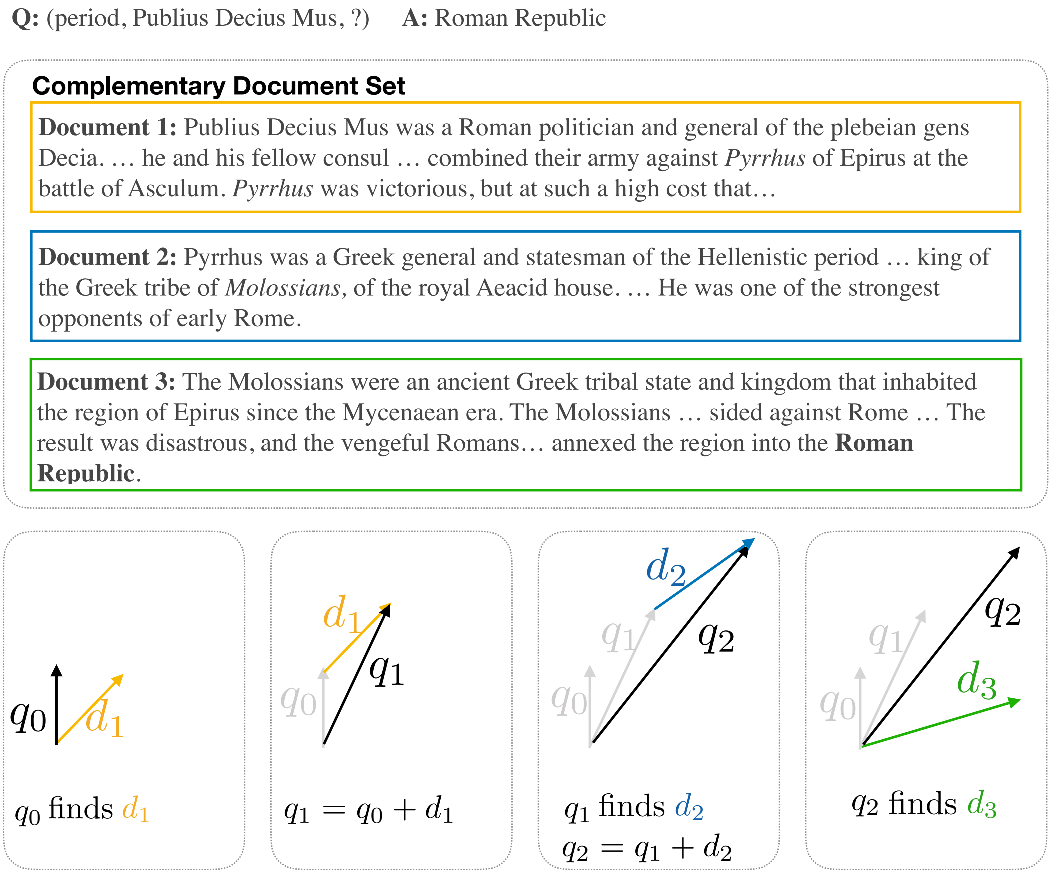
When considering the task of retrieving relevant documents to a given search query, *Relevance Feedback* (Rocchio, 1971) has been developed as a local analysis technique to improve retrieval performance. It takes into account user feedback on the relevancy of retrieved documents, and updates search results with user-provided relevancy information. In *Rocchio's algorithm*, the initial query vector is shifted towards the centroid of documents annotated as relevant, and away from the centroid of irrelevant documents. This implicit expansion of the search query with terms from relevant documents can help to better contrast relevant from irrelevant documents by incorporating new and distinctive terms into the query, which can translate into substantial retrieval improvements. However since in practice user annotations are not always available, *Pseudo Relevance Feedback* (PRF) (Salton and Buckley, 1997) presents a viable alternative. PRF considers the highest-ranking documents obtained via an initial query as (pseudo-)relevant, and uses these instead of actual user relevance feedback to augment the initial query vector. At the cost of potential semantic drift when erroneously retrieved irrelevant documents are included, retrieval performance often improves, due to overcoming the problem of extreme sparsity in TF-IDF query vectors.

Previous work on applying RC systems in conjunction with larger text collections has mostly used uni- or bigram TF-IDF (or variations thereof) as retrieval component (Chen et al., 2017a; Yang et al., 2018; Thorne and Vlachos, 2019). PRF differs from TF-IDF in that it scores retrieved documents not only using the query, but also using the *content* of other retrieved documents. It is this property that makes PRF potentially better suited for selecting combinations of documents in WIKIHOP: PRF determines the relevance of documents to a query partly co-dependent on the content of other documents. The PRF query vector could thus be updated using document terms that also include the bridge entities mentioned in separate documents, and help recover chains of relevant documents. We will next illustrate this in an ideal example.

5.3.2 Illustration: Multi-Step Retrieval with PRF

In Figure 5.2 we schematically illustrate how iteratively applying PRF can change the query vector to retrieve the document mentioning the answer. An initial query vector q_0 about the era or period that *Publius Decius Mus* lived in is generated from the query. It serves to retrieve document d_1: the WIKIPEDIA article about *Publius Decius Mus* (yellow), i.e. the subject entity mentioned in the query. Unfortunately (though by task design) this document d_1 does not mention the correct answer (*Roman Republic*), and is thus insufficient for an extractive RC model.

But PRF considers this retrieved document d_1 as relevant, and uses the terms within d_1 to form a new query vector q_1. Compared to the first query q_0, the vector



Q: (period, Publius Decius Mus, ?) **A:** Roman Republic

Complementary Document Set

Document 1: Publius Decius Mus was a Roman politician and general of the plebeian gens Decia. … he and his fellow consul … combined their army against *Pyrrhus* of Epirus at the battle of Asculum. *Pyrrhus* was victorious, but at such a high cost that…

Document 2: Pyrrhus was a Greek general and statesman of the Hellenistic period … king of the Greek tribe of *Molossians,* of the royal Aeacid house. … He was one of the strongest opponents of early Rome.

Document 3: The Molossians were an ancient Greek tribal state and kingdom that inhabited the region of Epirus since the Mycenaean era. The Molossians … sided against Rome … The result was disastrous, and the vengeful Romans… annexed the region into the **Roman Republic**.

Figure 5.2: An illustration for the use of recursive Pseudo-Relevance Feedback for multi-hop document selection. An initial query vector is recursively updated based on the terms in previously retrieved documents.

q_1 is shifted towards terms from this first retrieved document d_1 (yellow), also including the term *Pyrrhus*. When then applying TF-IDF search again, q_1 now also retrieves a second document d_2: the WIKIPEDIA article about *Pyrrhus* (blue), yet the correct answer a is still not mentioned in this document. Applying PRF once more based on q_1 creates yet another search vector q_2, which is further shifted towards the terms in d_2, now including the term *Molossians*. This second refined query vector q_2 can then retrieve a document d_3 which mentions the *Molossians*, alongside the correct answer *Roman Republic*. That is, by recursively applying PRF (twice), the initial query vector has been gradually updated and shifted towards other terms in such a way that it enables the retrieval of more documents – including those mentioning the correct answer. In doing so, PRF will rank highest those documents mentioning the most specific terms from both the query and the initially retrieved documents (i.e. terms with high IDF score).

The example demonstrates that recursively applying PRF can potentially be a useful method for finding relevant documents in WIKIHOP. But the example is an ideal one, and it is unclear whether PRF can help retrieve such relevant documents in practice. PRF is by no means precise; there is a risk of semantic drift; in addition, irrelevant document chains, e.g. those leading to distractor candidates will *also* be retrieved, as they are linked to the root documents in the same way as documents mentioning the correct answer. Thus, in the next section we will quantitatively explore the use of PRF for selecting documents in WIKIHOP, comparing it to several other retrieval methods.

5.3.3 Experiment: Comparison of Retrieval Methods

In this section we will conduct experiments with the aim of measuring the extent to which different retrieval methods can identify relevant documents to answer a given multi-hop query on WIKIHOP. Besides PRF we will consider a variety of other retrieval mechanisms for document selection.

First, we test the commonly used TF-IDF and BM25 retrieval methods, each both for unigrams and for bigrams. These retrieval approaches measure relevance in terms of lexical overlap with an individual document, weighted by term specificity. As a second type of baseline, we will test thesaurus-based query expansion (QE), where the query is augmented with additional synonyms to query tokens from WordNet (Miller, 1995), as well as from *thesaurus.com*. Third, we consider query expansion based on an automatically constructed thesaurus, where we select the nearest neighbours[3] of pre-trained *GloVe* (Pennington et al., 2014) word embeddings for each query token and append these to the query, tuning the number of neighbouring terms in $\{1,2,5,10\}$. Fourth we will test PRF based on origial TF-IDF queries, and tune the number of documents used for relevance feedback in $\{1,2,5,10\}$. Finally, we examine recursive PRF with two recursions, again tuning the number of documents used for relevance feedback in each level of recursion in $\{1,2,5,10\}$.

[3] We use *cosine* similarity to rank word pairs and find nearest neighbours.

		Hits@k		
Retrieval Method	**MRR**	1	3	10
TF-IDF (unigram)	25.6	8.9	41.3	50.6
BM25 (unigram)	24.3	6.5	41.5	50.6
TF-IDF (bigram)	25.8	11.1	39.2	48.1
BM25 (bigram)	24.8	8.9	40.2	48.3
QE – Thesaurus	32.3	12.7	48.7	66.4
QE – Automatic Thesaurus	39.4	18.7	53.1	81.7
PRF	42.4	19.2	57.7	90.0
Recursive PRF	**45.9**	**25.0**	**59.0**	**90.1**

Table 5.1: Comparison of different retrieval methods for WikiHop, results in [%]. QE: Query Expansion, PRF: Pseudo Relevance Feedback.

These baselines allow for a comparison of different contributing factors: we can measure the effect of using different types of query expansion (local vs. global), or not using query expansion at all. While the thesaurus and automatic thesaurus-based query expansion strategies provide general additional information about the query tokens, PRF uses the local information from some initially retrieved documents. It is worth pointing out that both PRF and the other QE baselines generally reduce the sparsity of query vectors. By comparing PRF with other QE baselines, we can then measure the extent to which potential improvements are due to sparsity reduction, compared to the inclusion of information from several documents, which is specific to PRF. Finally, the usage of pre-trained word embeddings in the automatic thesaurus baseline can give an indication for the merits of distributional information of the query words for determining document relevancy.

The previously described retrieval methods each produces a ranked list of individual documents. For evaluation, we identify the rank of the first document which mentions the correct answer, and use it to calculate Hits@k and Mean Reciprocal Rank (MRR) scores. The results are summarised in Table 5.1.

First, we observe that TF-IDF and BM25 produce a very similar outcome, both for the unigram and for the bigram case. Compared to all other modifications tested, making changes here does not lead to major changes in performance. Next, we observe clear improvements over standard TF-IDF from using *any* type of query expansion, i.e. both from local and from global information. This is plausible, since

WikiHop query vectors are very sparse, and query expansion of any type can help overcome this sparsity. Furthermore we observe that global information is beneficial both when using a curated as well as an automatically induced thesaurus. In particular the automatic-thesaurus expansion, which relies on the similarity of word embeddings, shows substantial improvements over both TF-IDF, BM25 and the synonyms from *thesaurus.com*. That is, distributional information on the entity in the query is an important cue for identifying relevant documents that mention the answer: query terms and terms from these retrieved documents tend to share the same distributional context, which is picked up and reflected in the respective *GloVe* vectors.

The local query expansion approaches improve retrieval performance even further. Compared to standard TF-IDF, PRF improves document selection by a substantial margin; recurring PRF twice then almost triples the Hits@1 score compared to unigram TF-IDF. That is, by using PRF we can improve the selection of relevant documents for WikiHop, and to an extent that goes beyond the improvements that we witness with other, global methods for query expansion, which equally help overcome the problem of query sparsity. Finally we see that these improvements are largest when applying PRF a second time.

In summary, this experiment shows that (recurred) PRF can indeed provide a potent means to improve document selection on WikiHop when compared to methods considering the relevance of documents in isolation, such as TF-IDF and BM25. The benefits of this go beyond those conferred by reducing query vector sparsity, since PRF outperforms other query expansion approaches. Having established this, we will next investigate a model pipeline approach where – relying on PRF – we combine the selection of document tuples with neural RC models.

5.4 Document Combinations as a Latent Variable

So far we have observed that PRF presents a viable starting point for selecting individual documents mentioning the correct answer for WikiHop queries – much more so than TF-IDF. But how can this observation be leveraged in an integrated system that uses the outputs of selected *combinations* of documents for an RC model to then process further? We will next lay out a broad structure for such a pipeline system. This framework will later also serve as the groundwork for comparing different document selection methods in conjunction with different RC models.

Recall that in the WIKIHOP task, our goal is to find the answer a to a query q about a set of given documents S. To model this problem, we will introduce a new latent variable: we will consider tuples $D = (d_1, \dots, d_T)$ of size T among the given support documents S, where $T \in \mathbb{N}$. The full space of arbitrary possible tuples D is thus S^T, the T-fold Cartesian product over S. Having introduced this new latent variable, we can rewrite the probability for predicting the correct answer by marginalising over $D \in S^T$:

$$\begin{aligned} P(a \mid S, q) &= \sum_{D \in S^T} P(a, D \mid S, q) && (5.1) \\ &= \sum_{D \in S^T} P(D \mid S, q) \cdot P(a \mid D, q) && (5.2) \\ &= \mathbb{E}_{D \sim P(. \mid S, q)} [P(a \mid D, q)] && (5.3) \end{aligned}$$

where we rely on the definitions of conditional probability and expectation for Equations 5.2 and 5.3, respectively, and furthermore drop the dependence of the answer probability distribution $P(a \mid D, q)$ on S in Eq. (5.2). That is, we introduce a variable D for a particular combination of documents, and we use it to factor the answer prediction probability $P(a \mid S, q)$ into one factor $P(D \mid S, q)$ for document combination selection and one factor $P(a \mid D, q)$ for an RC model probability, both of which are then aggregated across all possible options of D in S^T. This effectively decomposes the problem into i) selecting a particular document tuple D to read, ii) predicting an RC probability for the correct answer a, conditioned on only the query and D, rather than q and S, and iii) the aggregation of results across different values for D.

The advantage of this factorisation is that it separates text combination selection and RC module, allowing for a direct comparison of different choices for each.

Furthermore, the RC model is now conditioned on smaller subsets of documents in S, providing it with less text to read than the full set of all documents S, which is potentially relevant considering that RC models can be distracted when given irrelevant portions of text (Jia and Liang, 2017).

5.4.1 Pruning the Document Combination Space via Recursive PRF

The full space of possible combinations, i.e. tuples $D = (d_1, \ldots, d_T) \in S^T$ grows exponentially with respect to the tuple size T. As a consequence, even for small values of T, it becomes impractical to fully cover the space of all possible document combinations as inputs to a computationally costly RC model or text relevancy model in the inner loop during model training. In order to overcome this problem we will have to restrict the number of available document combinations. We will thus prune S^T and consider only a restricted set of document tuples, and we will use recursive PRF to achieve this. From the initial query q and given support documents S, we will construct a sparse document graph G, in which paths correspond to document combinations. This graph $\mathrm{G} = (S, E)$ connects documents from S via a set of edges E by following the document chains obtained when recursively applying PRF. This process will now be described in detail.

An initial set of seed documents $S_0 \subseteq S$ is obtained via standard TF-IDF retrieval, based on an initial search query $q_0 = q$. This query is subsequently expanded to q_1 with PRF based on S_0. We then query S again with the search vector of q_1 and obtain the next set of retrieved documents S_1. This process is repeated, and after having recursively applied PRF for $T-1$ times, we have obtained a sequence of search query vectors $q_0, q_1, \ldots, q_{T-1}$, each with its own respective sets of retrieved documents $S_0, S_1, \ldots, S_{T-1}$. The connectivity of the document graph G is then defined as follows: two document nodes (d, d'), with $d \neq d'$ are connected iff $\exists\ t \in \{0, 1, ..., T-1\}$ such that $d \in S_t \wedge d' \in S_{t+1}$; that is, when d and d' appear in two subsequently retrieved sets of documents S_t and S_{t+1}.

If G were fully connected, then the set of paths of length T in the graph would correspond to *all* document combinations S^T. However, when using only the paths defined by recursive application of PRF, the resulting graph will generally be sparse. The sparsity of G can further be controlled by restricting the maximum size of each document set S_i, or alternatively by setting a lower threshold for a minimum required retrieval score threshold. In summary, we reduce the number of document combinations under consideration by restricting ourselves to combinations of documents retrieved in subsequent levels of recursive PRF.

Intuitively, two documents d and d' are then connected if d' contains the most specific words from either d or the query. The method can thus include document tuples where one document mentions an entity name that re-appears in the subsequent document of the tuple. Consequently, the mechanism can assemble complementary information about entities mentioned in separate documents along the paths of G, as required in the WIKIHOP task.

5.4.2 A Model for Document Combination Probabilities

Even though we have established that PRF compares favourably to TF-IDF and other retrieval approaches when selecting relevant documents, PRF is still not precise, and can in absolute terms not be expected to reliably provide relevant input to an RC model. In order to improve the relevancy of the selected document combinations further we will thus additionally score the paths in G with a model that learns a probability $P(D \mid S, q)$ for the relevance of a particular document combination D. That is, PRF is merely used as a first step to identify the inputs (document paths) for a subsequent document path ranker, which learns a probability distribution over potentially relevant document combinations. A good model can then ideally learn to assign lower scores to combinations irrelevant to the original query, and furthermore circumvent potential negative consequences of semantic drift in PRF.

From here on we will refer to the set of all paths of length T in the graph G as $\mathrm{D}_T(S,q)$. Concretely, for the model proposed in Equation 5.2, the restriction to paths in the graph means that rather than summing over *all* $D \in S^T$, we will restrict the sum to only $D \in \mathrm{D}_T(S,q)$. We will next introduce a document combination scoring model for $D \in \mathrm{D}_T(S,q)$, with the aim of learning which document combinations are relevant to a query.

The model assigns a probability $P(D \mid S, q)$ to a combination of documents $D \in \mathrm{D}_T(S,q)$, given the query and S, and is normalised over $\mathrm{D}_T(S,q)$. We choose a model that learns to contrast positive (relevant) combinations from negative (random) combinations in a binary classification task using a form of negative sampling. The resulting binary prediction score achieved for any one given $D \in \mathrm{D}_T(S,q)$ is then interpreted as the logit for a softmax distribution over all $\mathrm{D}_T(S,q)$. More concretely, we sample positive training examples D uniformly at random from all paths $D \in \mathrm{D}_T(S,q)$ in G which contain both a document mentioning the correct answer a_i, as well as the question entity. To contrast against these positive samples, we sample negative examples in equal proportion uniformly at random from all paths $\mathrm{D}_T(S,q)$. We then train a binary classification model with a cross-entropy minimisation objective to distinguish positives samples from negatives. We train the binary classifier on features consisting in the element-wise *min*, *mean*, and *max* of pre-trained CBOW embeddings (Mikolov et al., 2013a), pooled across tokens, computed individually for both the query q and each document d of the combination D, and finally concatenated into one vector representation. Simple aggregate feature representations have been shown to be effective in text classification (Joulin et al., 2017), and are in addition very fast to compute as they can be pre-computed for each document and query. Our classification model is an ensemble of 200 decision trees (Breiman et al., 1984; Breiman, 2001), restricted to a maximum depth of 40.

This overall classification model combines the following advantages: i) fast, and potentially parallelisable inference due to the parallel nature of the tree ensemble model ii) potential to pre-compute and re-use feature vectors iii) like a two-layer MLP, the Random Forest model is endowed with the full modelling capacities of a nonlinear function approximator, allowing for the representation of arbitrary interactions between query and document features. In summary, we have described an efficient model for learning a probability distribution $P(D \mid S, q)$ over document combinations $D \in \mathrm{D}_T(S,q)$.

At inference time we score only those document combinations D retained after pruning S^T with recursive PRF, rather than the full space S^T, and weigh respective

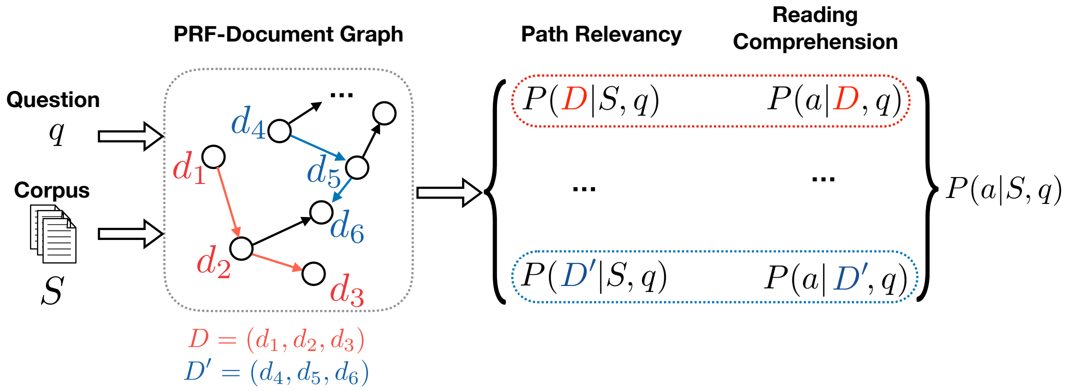


Figure 5.3: Method Overview: A question q and corpus S are used to induce a document graph via recursive Pseudo Relevance Feedback. Individual paths (such as the ones shown in red and blue) represent potentially relevant complementary evidence. Answer probabilities are computed conditioned on different paths, combined with answer prediction probabilities, and finally marginialised over different paths.

answer predictions of an RC model according to $P(D \mid S,q)$:

$$P(a \mid S,q) = \sum_{D \in \mathrm{D}_T(S,q)} P(D \mid S,q) \cdot P(a \mid D,q) \tag{5.4}$$

That is, both document combination relevance score and RC answer score are aggregated into a final model prediction by aggregating model probabilities over $\mathrm{D}_T(S,q)$. Figure 5.3 illustrates this overall system structure in a high-level overview.

5.4.3 Combination with Text Comprehension Models

When training the reader component $P(a \mid D,q)$, we sample uniformly, both from the training set (indexed by i) and then uniformly among the corresponding $D \in \mathrm{D}_T(q_i,S_i)$ in which one document mentions the correct answer a_i, and one document mentions the entity in the question. Tuning the RC model then amounts to standard cross-entropy training to compute the correct answer. While there is a mismatch to training the RC model parameters directly under the expectation with respect to $P(D|S,q)$ (cf. Equation 5.3), the training distribution used here ensures that during RC training the correct answer can always be found within the documents, and that the document about the entity in question is also always included. The RC model is thus optimised on a distribution of document combinations for which the text selection model is trained to assign high probabilities.

To summarise all these individual model components, we first assemble a sparse document graph by recursively applying PRF; subsequently we score individual paths in this graph using a text selection model that we train with negative sampling, and with an RC model to predict answers; finally all respective scores are aggregated according to Equation 5.4 to form a final answer prediction probability. Having described this full model, we will next evaluate it experimentally.

5.5 Experiments: WIKIHOP

We will now evaluate the above proposed method on WIKIHOP using two different neural RC models, *FastQA* (Weissenborn et al., 2017) and *BiDAF* (Seo et al., 2017a). As comparison baselines, we also form pipeline models based on TF-IDF alone, and with PRF but without the additional trainable ranker, each similarly combined with one of the two neural RC models.

For the TF-IDF-based selection approach, we first issue the query q to retrieve a ranked list of documents in S and use these to assemble document tuples of size T. Tuples are ordered by the maximum TF-IDF rank that any constituent document in the tuple has; in the end we retain the top K tuples. We tune K in $\{1,2,5,10\}$ and aggregate RC predictions with equal contribution to form the final prediction for the answer a.

In a second baseline PRF is used for selecting document tuples, but without the trained model for document combination probabilities. We assemble a list of document tuples as follows: we conduct recursive PRF, retrieving the sets S_0, S_1, $\ldots$, S_{T-1} which are also used to define G. Next we fill each tuple position with a document in $S_0, S_1, ..., S_{T-1}$, respectively. Different tuples are then sorted according to the maximum retrieval rank that any document d_t has within its respective set S_t, aggregated across all tuple positions t. Again, we retain only the top-K ranked document combinations, tuning K in $\{1,2,5,10\}$.

In summary, these two baselines approaches consist in pipeline models that –

	FastQA	*BiDAF*
TF-IDF + reader	30.1	32.8
PRF + reader	37.5	44.6
Trained ranker + reader	**54.1**	**57.7**

Table 5.2: WIKIHOP accuracy for different choices of document combination selection (TF-IDF, PRF, or a trained document combination *ranker* model), and neural *reader* (either *FastQA* or *BiDAF*).

rather than using the trained document combination ranker – perform text selection either based on TF-IDF or based on PRF. We select $T = 3$, train the models on the WIKIHOP training set, keeping aside 5% for tuning purposes, and evaluate in the standard (unmasked) validation set. The RC models are furthermore restricted to only predict answer candidates, rather than *any* span in the document.

5.5.1 Experimental Outcome

Do the retrieval improvements observed for PRF compared to TF-IDF in Table 5.1 also translate into better downstream answer prediction accuracy? The results of this experiment can be found in Table 5.2, both for *FastQA* and *BiDAF* as RC model component. Indeed, using recursive PRF for document selection leads to a substantial improvement over TF-IDF: 7.4% better accuracy for *FastQA* and 11.8% for *BiDAF*. This again underscores the suitability of PRF for document selection in the WIKIHOP task.

Including the trainable document tuple ranker then further improves downstream accuracy, and by a substantial margin: an additional 16.6% for *FastQA*, and 13.1% for *BiDAF*. Furthermore, when evaluating the document tuple prediction model in terms of pure retrieval success, it reaches a 66.9% Hits@1 score (among document triples), which – although an imperfect juxtaposition – compares favourably to the Hits@3 score of 59.0% measured for individual documents with recursive PRF in Table 5.1. That is, the additional step of learning a relevance distribution for document combinations improves the results beyond those of PRF and TF-IDF, both in terms of pure retrieval and also in terms of downstream accuracy.

In summary, we have shown that recursive PRF does not only provide more relevant document tuples than TF-IDF, it can also serve as a pruning method to restrict the number of tuples that are to be evaluated in the marginalisation step of a latent variable model, which outperforms both pure TF-IDF and recursive PRF by a large margin.

5.6 Application of the Method on HOTPOTQA

The PRF-based document tuple ranking method described so far in this chapter was developed with the aim of improving model performance on the WIKIHOP dataset. But how does this approach fare on a different multi-hop dataset, constructed with a different dataset induction paradigm? We next conduct experiments on the HOTPOTQA dataset (Yang et al., 2018), a dataset also assembled with the aim of fostering multi-hop Reading Comprehension. Like WIKIHOP, HOTPOTQA is constructed based on WIKIPEDIA articles, but rather than posing structured queries of the form `(entity, relation, ?)` which seek answer entities, it poses natural language questions composed by crowd annotators. In both datasets, answers are mentioned as spans in the texts (disregarding the yes/no questions in HOTPOTQA for now). Both WIKIHOP and HOTPOTQA furthermore contain a significant amount of questions that involve texts connected by *bridge* entities. Given these structural similarities and differences between the two datasets, how well do different retrieval methods, as well as the above described pipeline approach fare on HOTPOTQA?

5.6.1 Comparison of Retrieval Methods

We first compare several retrieval baselines on the *bridge* setting in the *dev-distractor* part of the HOTPOTQA dataset. In this setting, a small set of relevant documents is pre-selected using a variation of TF-IDF (Yang et al., 2018), and combined with documents containing the relevant information. This *bridge* setting more closely resembles the type of entity-based cross-document hopping seen in WIKIHOP (where one would expect PRF to potentially be useful), in contrast to the *comparison* portion of the dataset, which focuses on yes/no questions without

		Hits@*k*		
Retrieval Method	**MRR**	2	3	10
TF-IDF (unigram)	11.2	6.1	11.3	38.8
BM25 (unigram)	12.7	8.9	14.9	41.5
TF-IDF (bigram)	10.9	6.4	11.1	36.4
BM25 (bigram)	11.6	7.8	13.1	37.7
QE – Thesaurus	7.5	4.3	8.3	33.5
QE – Automatic Thesaurus	8.5	5.6	10.5	36.2
PRF	**13.2**	**9.4**	**15.9**	43.2
Recursive PRF	**13.2**	**9.4**	15.8	**43.3**

Table 5.3: Comparison of different retrieval methods for HOTPOTQA, results in [%]. QE: Query Expansion, PRF: Pseudo Relevance Feedback.

requiring an entity bridging structure.

In HOTPOTQA we furthermore have access to human annotations on the sentence level for answering the question. Rather than selecting full documents, we thus perform retrieval and evaluation on the level of *individual* sentences of the given documents. For evaluation, we then measure the first rank at which *all* sentences annotated as relevant have been retrieved, comprehensively. Since always a minimum of two sentences are labelled as necessary in HOTPOTQA, we calculate *Hits@k* statistics, beginning at $k = 2$.

Results In Table 5.3 we present the outcome of this experiment. For HOTPOTQA we observe that PRF – both in its standard and recurred variant – outperforms TF-IDF, BM25, and all other approaches, albeit only by a small margin: by far not to the same extent as previously observed for WIKIHOP. Interestingly, QE based on global information is less useful on HOTPOTQA: both thesaurus- *and* automatic thesaurus-induced word embedding similarity actively deteriorates performance compared to standard TF-IDF and BM25. This is noteworthy, especially in contrast to the results of Table 5.1: the benefits reaped from using QE on WIKIHOP queries do largely not transfer to HOTPOTQA. A possible explanation is the inherent sparsity of the query vectors in WIKIHOP – which is partly responsible for the improvements of using QE on WIKIHOP – whereas for HOTPOTQA sparsity is not as big an issue. PRF and recursive PRF in this setting show modest improvements over the baselines without QE, yet thesaurus-based augmentation appears to

	Br EM	Br F_1	Cp EM	Cp F_1
TF-IDF* + *reader* (Yang et al., 2018)	**19.76**	**30.42**	**43.87**	**50.70**
trained *ranker* + *reader*	15.44	22.39	40.42	47.64

Table 5.4: HOTPOTQA Full Wiki results for Bridge (Br) and Comparison (Cp) questions. The results for the first row stem from the original publication (Yang et al., 2018), where more details on the implementation of TF-IDF can be found, which slightly differs from standard TF-IDF.

actively distract the document selection from finding relevant text.

5.6.2 Comparison on the QA Task

If PRF is less useful on HOTPOTQA than on WIKIHOP, how does this affect the above developed pipeline system on HOTPOTQA? We next test the system using the RC component described in the original paper (Yang et al., 2018), which amalgamates several commonly used neural components of recent RC architectures.[4] We combine the RC model with the trainable ranker, set $T = 2$ and use document pairs containing the human-annotated relevant sentences as positive samples for relevancy training. In Table 5.4 we compare the results of this model with the originally reported results for the *full-wiki* dev setting (Yang et al., 2018), where a variation of TF-IDF is used for selecting relevant documents.

We observe that our above developed pipeline system achieves overall comparable, but clearly lower exact match (EM) and answer overlap (F_1) scores compared to the originally reported pipeline of TF-IDF and RC model (Yang et al., 2018). These results are consistent across metrics, and also when considering results individually for either *bridge* or *comparison*-type questions. That is, relying on PRF rather than TF-IDF for document selection on HOTPOTQA does not translate into downstream answering improvements, as we had observed for WIKIHOP.

5.6.3 Discussion

What can explain the gap in the usefulness of PRF between the two datasets? Both are designed to pose multi-hop questions; they share the same WIKIPEDIA domain, and the answers (at least for the bridge questions of HOTPOTQA) are often entities,

[4] https://github.com/hotpotqa/hotpot

	HOTPOTQA-Short	**HOTPOTQA-Long**
TF-IDF* + *reader*	11.63	**19.77**
trained *ranker* + *reader*	**13.95**	15.46

Table 5.5: HOTPOTQA: Comparison of EM (in [%]) on short and long bridge questions for document selection via TF-IDF*, and via the trained *ranker*.

just like in WIKIHOP.

One possible explanation for the differences in the usefulness of PRF that we observe between the two datasets is the different type of query structure. Where in WIKIHOP we have structured KB queries, in HOTPOTQA the questions are natural language expressions composed by human annotators. Where in WIKIHOP the texts are selected based on the queries using a noisy distant supervision assumption, in HOTPOTQA the questions are composed *given* the texts, thus guaranteeing the relevance of the text for the question, but also increasing the likelihood of lexical transfer between these given (relevant) texts into the question.

Consequently, HOTPOTQA has on average much longer queries than those in WIKIHOP: the average length differs by 12 tokens between the datasets. A first possible explanation might then be that the benefits of using PRF (and perhaps query expansion more generally) stem from overcoming the inherent information scarcity of the shorter queries in WIKIHOP, i.e. from performing QE in the first place.

A further analysis of model predictions on HOTPOTQA confirms that question length plays a role: when we break down results on the *Bridge* questions (those excluding yes/no questions) in Table 5.5, we find that the usefulness of performing QE on short HOTPOTQA questions is very different. We observe that questions with 10 or more tokens (“long”), which make up the majority of the questions, are better answered using basic retrieval with no query expansion. Short questions with less than 10 tokens, on the other hand, are more easily answered relying on PRF. This adds empirical weight to the explanation that one of the ways in which query expansion confers its benefits is by reducing the relative sparsity of the query vector in short questions, as we had hypothesised for the case of WIKIHOP.

It is worth noting that for TF-IDF-based retrieval we observe a marked performance increase when shifting from short (11.63% EM) to long questions (19.77% EM). A possible explanation for this is that longer questions have more lexical overlap with the texts they are composed about, making them more suitable for retrieval with TF-IDF. In fact, we also observe that for long questions, ngram-overlap in particular starts to become a very strong retrieval signal: in a small ablation study we test retrieval by largest common ngram. This method ranks documents according to the length of the longest common ngram between question and document. We find that ngram overlap alone is a very strong indicator for the relevance of a document, though interestingly also dependent on the question length: while for short questions a relevant document is retrieved in only 4.65% of cases (Hits@1), for questions with 10 or more tokens this number more than sextuples to 28.15% Hits@1.

The usefulness of ngram overlap for the retrieval of long queries, but not for short queries, suggests that in the crowdsourced data annotation method employed in HOTPOTQA, question annotators directly lift over sub-sequences from their given text into the question, and in particular when they compose long questions. This is a second explanation for the usefulness of the lexical overlap-based TF-IDF retrieval in HOTPOTQA. When employing query expansion of any sort on long questions, we add new tokens into the query vector, diluting the relatively clear signal already provided by the re-use of terms from the text in the original question. For shorter questions on the other hand, where both ngram-overlap is less indicative *and* query vectors are sparser, PRF-based expansion is more useful than TF-IDF.

In conclusion, we have isolated two factors that are suggested by these experiments as plausible explanations for the difference in the usefulness of PRF between WIKIHOP and HOTPOTQA: the first being a reduction of query vector sparsity for the shorter queries in WIKIHOP as well as the short questions in HOTPOTQA; and second the usefulness of direct lexical overlap strategies on longer HOTPOTQA questions stemming from the direct transfer of relevant text sub-sequences into the question.

5.7 Conclusion

In this chapter we have investigated the use of different retrieval methods and query expansion strategies for selecting relevant documents in a cross-document RC task. We can thus summarise the answers to the research questions posed in the beginning of this chapter as follows:

1. **How does PRF compare to other IR methods when retrieving relevant documents on WIKIHOP?** We have empirically demonstrated that, for WIKIHOP, PRF compares favourably to other approaches which rank documents independently – namely unigram and bigram TF-IDF and BM25, thesaurus-based query expansion, and query expansion based on word vector similarity. Compared to TF-IDF, which is widely used for text selection in RC contexts, recursively applying PRF leads to an improvement from 8.9 to 25.0 Hits@1.

2. **How can PRF be adapted to retrieve document combinations, rather than individual documents?** We have described a method of using PRF to retrieve combinations of documents which gradually builds up document tuples based on subsequently retrieved sets of documents when recursively applying PRF.

3. **How can document combinations retrieved with PRF be integrated with a trainable ranking model and neural RC system, and how does this translate into downstream answer prediction performance on WIKIHOP?** Document combinations can be viewed as a latent variable, and PRF-based selection can be used to prune an otherwise exponentially large marginalisation space. When combined with with a neural RC system, this approach compares favourably to TF-IDF-based document selection; adding a trainable document combination ranking model further improves answer prediction performance on WIKIHOP.

4. **How do observations on document selection for WIKIHOP compare with observations for HOTPOTQA; can they be related to different dataset in-**

duction choices? The improvements on WIKIHOP observed with the use of PRF, rather than TF-IDF, do largely not translate to HOTPOTQA, neither for text selection alone, nor when measuring downstream QA performance. We related the observed discrepancy to elementary differences in query properties (length, lexical overlap) of these two datasets, which are ultimately rooted in their different dataset induction strategies.

Part III

Machine Comprehension Model Undersensitivity

Chapter 6

Exploring Undersensitivity of Neural Reading Comprehension Models

The content of this chapter is based on unpublished work (Welbl et al., 2020b). For further context, on the OpenReview website[1] critical reviews can be found, which have subsequently been taken into account. The experiments in Table 6.9, data preparations for Table 6.10 and data preprocessing for Table 6.6 were conducted by collaborators in Welbl et al. (2020b).

6.1 Introduction

We have in the previous Parts I and II of this thesis considered RC datasets for both science exam questions and multi-hop inference, alongside a series of different models. Next we focus with more detail on individual comprehension questions and models' behaviour on an established task. Do RC models adequately represent and take into account all relevant information specified in a given comprehension question? Do they identify answers for the *particular* information request formulated, or do they rely on shortcuts and shallow predictive cues that help them answer the question correctly, but could bring the model to fail when exploited? We will now adopt an adversarial perspective on RC models and explore questions on model behaviour and their sensitivity (or lack thereof) to modifying the model input. As an

[1] `https://openreview.net/forum?id=HkgxheBFDS`

RC model's behaviour depends on the data it is trained on, we will relate the model failures we identify to particular properties of the training set. Concretely, we will demonstrate that an RC model's lack of sensitivity to meaningful changes in its input question are due to a lack of structurally similar, but unanswerable questions in the RC dataset.

6.1.1 Adversarial Vulnerability and Undersensitivity in NLP

Neural networks – the core ingredient of contemporary RC models – have been shown to be vulnerable to adversarial perturbations of their input (Szegedy et al., 2013; Kurakin et al., 2016). More concretely in NLP, which operates on discrete symbol sequences, adversarial examples have been studied extensively – see (Zhang et al., 2019) for a recent survey. Adversarial attacks can take a variety of forms (Ettinger et al., 2017; Alzantot et al., 2018) including character perturbations (Ebrahimi et al., 2018), syntactic and lexical transformations (Li et al., 2017), semantically invariant reformulations (Ribeiro et al., 2018b; Iyyer et al., 2018), or adversarially pre-pended trigger text (Wallace et al., 2019a). Other prior work concentrates on specific NLP tasks and identifies adversarial attacks: for Fact Checking (Thorne and Vlachos, 2019), Machine Translation (Belinkov and Bisk, 2018; Zhao et al., 2018) or notably RC (Jia and Liang, 2017; Wang and Bansal, 2018; Mudrakarta et al., 2018), where adversarially chosen text insertions can drastically deteriorate a model's performance.

All of these attacks demonstrate that RC models – despite strong generalisation on test sets following the training distribution – are still unreliable, and can fail in surprisingly simple ways. A model's inability to handle adversarially chosen input text puts into perspective otherwise impressive generalisation results for in-distribution test sets (Seo et al. (2017a); Yu et al. (2018); Devlin et al. (2019); *inter alia*) and constitutes an important caveat to conclusions drawn regarding a model's language understanding abilities.

While the semantically invariant text transformations used in the above listed prior work can remarkably alter a model's predictions – demonstrating their *oversensitivity* to such transformations – the converse problem of model *undersensitivity*

Given Text	[...] The Normans were famed for their martial spirit and eventually for their Christian piety, becoming exponents of the Catholic orthodoxy [...]
Question (orig.)	What religion were the Normans?
Prediction (orig.)	Catholic orthodoxy (78%)
Question (adv.)	IP and AM are most commonly defined by what type of proof system?
Prediction (adv.)	Catholic orthodoxy (84%)
Given Text	The nearby Spanish settlement of St. Augustine attacked Fort Caroline, and killed nearly all the French soldiers defending it. The Spanish renamed the fort San Mateo [...]
Question (orig)	What was Fort Caroline renamed to after the Spanish attack?
Prediction (orig.)	San Mateo (98%)
Question (adv.)	What was Robert Oppenheimer renamed to after the Spanish attack?
Prediction (adv.)	San Mateo (99%)

Table 6.1: Examples of model undersensitivity in a BERT comprehension model trained on SQuAD2.0. Undersensitivity is a lack of input specificity: given the same text to read but altering the question can retain the prediction of the original question while increasing prediction probability.

is equally troublesome: the meaning of a model's text input can often be drastically changed while still retaining the original prediction with high probability. Two such examples are shown in Table 6.1 for a BERT (Devlin et al., 2019) model trained on SQuAD2.0 (Rajpurkar et al., 2018).

In the first example, when given the original question *"What religion were the Normans?"*, the model answers the question correctly as *"Catholic orthodoxy"* with 78% confidence. However, searching among a larger set of other, unrelated questions (in this case: other SQuAD training questions), shows that the model can be tricked into predicting the same answer also for an entirely unrelated question asked about the same paragraph, and with even higher confidence: *"IP and AM are most commonly defined by what type of proof system?"* is again answered with the expression *"Catholic Orthodoxy"* and 84% confidence. Clearly this answer is not correct; the question is unrelated to the topic of the given Wikipedia article, unanswerable from this context, and the model should have chosen a *NoAnswer* prediction. Nevertheless, BERT retains its original prediction, even increasing its confidence. Note that this is despite the addition of unanswerable questions (with label *NoAnswer*) into the SQuAD2.0 training set; the model has been trained to be able to predict if a question is unanswerable.

This particular example presents a striking failure case of the BERT model, but the question was discovered by searching among a large set of (generally unrelated) questions from the SQuAD training set. Often there are no more than a few such cases per sample – if any. Consequently it is hard to derive concrete insights from such examples, both regarding the model, or its training data.

The question in the second example in Table 6.1 is derived using a different and more systematic approach: a named entity in the question has been replaced with a different one. Again, the model fails to reflect a meaningful change in its input question: we observe an increased model probability for the same answer *"San Mateo"*, when replacing the entity *"Fort Caroline"* with *"Robert Oppenheimer"*. That is, the model prediction does not change, despite removing an essential component of the information request formulated in the original question, and replacing it with content unrelated to the paragraph or original question. This suggests that the entity *"Fort Caroline"* in the question might not be adequately represented and taken into account in the selection of *"San Mateo"* as answer.

Both of these two cases are examples of model *undersensitivity*: the model input is altered in a meaningful way – in such a way that a change of the model prediction would be adequate – yet the model prediction remains invariant. This stands in contrast to undesirable *oversensitivity* behaviour, where a model changes its prediction when it should not. Models prone to oversensitivity flip their prediction when applying a small, semantically invariant input transformation, e.g. replacing a token with a synonym (Ebrahimi et al., 2018). Undersensitivity is the opposite: a meaningful input change does not lead to a change in the model's prediction.

Excessive prediction invariance has been linked to adversarial vulnerability and led to impressive failure cases in computer vision (Jacobsen et al., 2019). Prior work on model undersensitivity in NLP (Feng et al., 2018; Ribeiro et al., 2018a) has shown that one can delete all but a small fraction of input words while models still produce the same output, and with a high probability. The inputs used in this prior work do however not constitute well-formed text, and are thus unnatural to a human reader. Consequently it is unclear which behaviour we should expect from natural language models evaluated on such unnatural text, and it is difficult to derive practical insights to improve RC models or datasets. By investigating well-formed and semantically coherent inputs – as we will in this chapter – we can demonstrate that undersensitivity is a phenomenon not limited to ill-formed expressions and partial text, but a phenomenon which extends to concrete and well-formed RC questions, where models provide false predictions.

6.1.2 Chapter Overview

In this chapter we will develop and test an automatic method for altering input questions, and use it to probe an RC model's undersensitivity behaviour. We formalise the process of finding such questions as an adversarial attack in a discrete input space arising from perturbations of the original question. There are two types of discrete perturbations that we consider – based on parts-of-speech, and on named entities – with the aim of obtaining grammatical and semantically consistent alternative questions that do not accidentally have the same correct answer.

Naturally, the success of an adversarial attack depends on the computational budget used to identify samples that satisfy the attack specification. We find that both SQuAD2.0 and NewsQA (Trischler et al., 2017) models can be attacked on a substantial proportion of samples, already with a small computational adversarial search budget, and the rates of successful attacks increase further as more input perturbations are considered. Observing a successful undersensitivity attack on a particular input sample is furthermore associated with lower standard performance metrics (EM/F_1), suggesting that the undersensitivity phenomenon – where present – is indeed a reflection of a model's lack of question comprehension, which can be exposed by probing it with input perturbations.

When training models to defend against undersensitivity attacks with data augmentation and adversarial training, we observe that they learn to generalise their robustness to held out evaluation data as well as held out perturbations – without sacrificing standard performance. Furthermore, we observe that the models trained with additional altered input data as 'negative' examples are also more robust in their generalisation to adversarial attacks defined in prior work (Jia and Liang, 2017), and show substantial improvements in a biased learning scenario with different training / evaluation set distributions.

These findings once more highlight the critical role of training data in RC: a lack of unanswerable counterexamples which are structurally similar to standard input samples is a cause of model undersensitivity. Including such samples into the training data alleviates the problem and considerably reduces the model's undersensitivity; at the same time it also improves the specificity in a model's question interpretation process. We summarise the research questions addressed in this chapter in the following list.

List of Research Questions:

1. How can RC model undersensitivity to changes in the question be evaluated using natural language inputs?

2. To what extent is the commonly used BERT (Devlin et al., 2019) model undersensitive to adversarially chosen input perturbations?

3. How can adversarially vulnerable samples be characterised and distinguished?

4. Can the two adversarial defence strategies of data augmentation and adversarial training help alleviate the undersensitivity problem?

5. How does training models to be more sensitive to input changes affect their behaviour?

6.2 Methodology

6.2.1 Problem Formalisation

We will begin by formalising our notion of model undersensitivity to question changes. Consider a discriminative model f_θ parameterised by a collection of dense vectors θ, which transforms an input x into a prediction $\hat{y} = f_\theta(x)$. In our task, the input $x = (d, q)$ consists of a given document d paired with a question q about d. The label $y(x)$ in our task is the answer to the comprehension question where it exists, or a *NoAnswer* label where it cannot be answered.[2]

In a text comprehension setting with a very large set of possible answers, predictions $\hat{y}$ should be *specific* to x, i.e. not the model prediction for arbitrary inputs. And indeed, randomly choosing a different input $x' = (d', q')$ is usually associated with a change of the model prediction $\hat{y}$. However, there exist many examples where the prediction erroneously remains stable; the goal of the attack formulated here is to identify such cases. Concretely, given a computational search budget, the goal is to discover inputs x', for which the model still erroneously predicts $f_\theta(x') = f_\theta(x)$, even though x' should be mapped onto a different prediction: the *NoAnswer* label of the RC task.

Identifying suitable candidates for alternative inputs x' can be achieved in manifold ways. A simple option is to search among a large question collection, as seen in the first example of Table 6.1, but we find this approach to only rarely be successful. Composing a new x' with a generative language model, on the other hand, is prone to result in ungrammatical or otherwise incoherent text. Instead, we consider a perturbation space $\mathcal{X}_{\mathcal{T}}(x)$ spanned by perturbing original inputs x using a perturbation function family $\mathcal{T}$:

$$\mathcal{X}_{\mathcal{T}}(x) = \{T_i(x) \mid T_i \in \mathcal{T}\} \tag{6.1}$$

This space $\mathcal{X}_{\mathcal{T}}(x)$ contains alternative model inputs derived from x that can be created by applying transformations in $\mathcal{T}$. Ideally the transformation function family $\mathcal{T}$ is chosen in such a way that they become unanswerable given d, i.e. for $x' \in \mathcal{X}_{\mathcal{T}}(x) : y(x') = NoAnswer$. The space $\mathcal{X}_{\mathcal{T}}(x)$ will later be used as a search space, and we will try to identify altered inputs $x' \in \mathcal{X}_{\mathcal{T}}(x)$ which erroneously retain the same prediction as x: $\hat{y}(x) = \hat{y}(x')$, even though the model should predict *NoAnswer*.

[2] Unanswerable questions are part of the SQUAD2.0 and NEWSQA datasets, but not SQUAD1.1.

6.2.2 The Part-of-Speech (PoS) Perturbation Space

We first consider a perturbation space $\mathcal{X}_{\mathcal{T}_P}(x)$ generated by PoS perturbations $\mathcal{T}_P$ of the original question. The perturbations $\mathcal{T}_P$ are defined by swapping individual tokens with other, PoS-consistent alternative tokens, where we draw from large collections of tokens of the same PoS types. Large collections of particular lexical realisations of a given PoS tag can be gathered automatically, e.g. from the RC training corpus. Any particular transformation changes one token; the result of applying one such perturbation $t_1 \in \mathcal{T}_P$ might then, for example, result in the following transformation, which exchanges the past tense verb *"patronised"* with *"betrayed"*.

$$\begin{aligned} q &: \textit{“Who \underline{patronised} the monks in Italy?”} \\ t_1(q) &: \textit{“Who \underline{betrayed} the monks in Italy?”} \end{aligned}$$

Individual token substitutions can furthermore be chained. For example, a second perturbation $t_2 \in \mathcal{T}_P$ might result in the following:

$$\begin{aligned} q &: \textit{“Who patronised the \underline{monks} in Italy?”} \\ t_2(q) &: \textit{“Who patronised the \underline{accountants} in Italy?”} \\ (t_2 \circ t_1)(q) &: \textit{“Who \underline{betrayed} the \underline{accountants} in Italy?”} \end{aligned}$$

Applying several individual transformations quickly results in questions that can be very different from the original. The set of possibilities grows exponentially with the number of perturbations applied, and the resulting question can be considerably different from the original and unanswerable given the document d.

There is generally no guarantee that the altered question will actually require a different answer (e.g. due to synonyms). Even more, there might be type clashes or other semantic inconsistencies (e.g. *"Who built the monks in Italy?"*). To avoid these, we found it useful to disregard perturbations of particular PoS types that frequently led to only minor changes or incorrectly formed expressions, such as punctuation or determiners. Concretely, we omit these following tags when perturbing questions: *"IN", "DT", ".", "VBD", "VBZ", "WP", "WRB", "WDT", "CC", "MD", "TO"*. Even then, the problem cannot entirely be circumvented. We will thus later conduct a qualitative analysis on adversarial attacks derived from this type of perturbation and quantify the issue of ill-formed questions. About half of the attacks generated based on this transformation resemble coherent and well-formed questions which do not accidentally still possess the same correct answer (cf. Section 6.4).

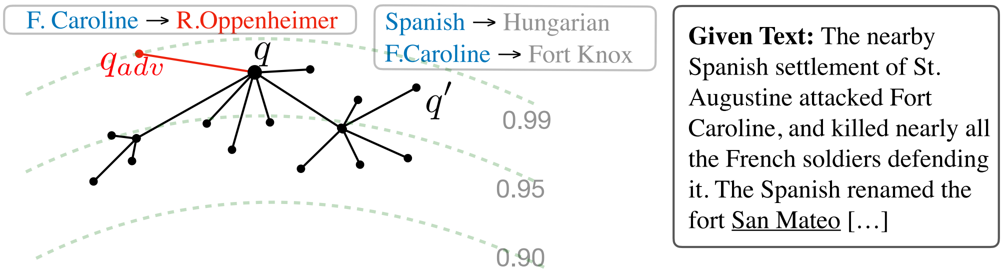
6.2.3 The Named Entity Perturbation Space

A different perturbation space $\mathcal{X}_{\mathcal{T}_E}(x)$ generated by the transformation family $\mathcal{T}_E$ is created by substituting mentions of named entities in the question with different type-consistent named entities. These replacements are drawn from a large collection of named entities (ordered by type), which can again be collected automatically from a sufficiently large text corpus. For example, a comprehension question *"Who patronised the monks in Italy?"* could be altered to *"Who patronised the monks in Las Vegas?"*, replacing the geopolitical named entity *"Italy"* with *"Las Vegas"*, chosen from a larger set of entities of this type. We observe that altering named entities often changes the specifics of the question while keeping it syntactically correct and semantically coherent. It furthermore specifies a different information request about a new, unrelated entity, which is unlikely to be satisfied from what is stated in the given document d, given the generally large number of possible named entities. While it is again not guaranteed that perturbed questions are in fact unanswerable or require a different answer, we will find in the later following qualitative analysis that in the large majority of cases they do (see Section 6.4).

6.2.4 Undersensitivity Attacks

Thus far we have described two different methods for perturbing an original question, each with a different space of perturbations. We will use the resulting perturbation spaces $\mathcal{X}_{\mathcal{T}_P}(x)$ and $\mathcal{X}_{\mathcal{T}_E}(x)$ to search for altered inputs x' for which the model prediction remains constant, compared to the original question: $\hat{y}(x) = \hat{y}(x')$. In addition we pose a slightly stricter requirement than only preserving the (argmax) prediction: f_θ should assign a higher probability to the same prediction $\hat{y}(x) = \hat{y}(x')$

Original Example (q):
What was *Fort Caroline* renamed to after the *Spanish* attack? San Mateo (0.98)



Given Text: The nearby Spanish settlement of St. Augustine attacked Fort Caroline, and killed nearly all the French soldiers defending it. The Spanish renamed the fort San Mateo […]

Adversarial Example (q_{adv}):
What was Robert Oppenheimer renamed to after the Spanish attack? San Mateo (0.99) ▲

Figure 6.1: Method Overview: Adversarial search over semantic variations of RC questions, producing unanswerable questions for which the model retains its predictions with even higher probability.

than for the original input:

$$P(\hat{y} \mid x') > P(\hat{y} \mid x) \tag{6.2}$$

That is, we search in a perturbation space for altered questions which result in a higher model probability for the same answer as the original question. Note that this is a conservative choice; it rules out cases in which the model might still produce the same prediction given the altered input, but with less certainty. If we can find an altered input question that satisfies the inequality (6.2), then we have identified a successful adversarial attack, and we will refer to it as an *undersensitivity attack*.

In its simplest form, a search for an adversarial attack in the previously defined attack spaces amounts to a search over a list of single lexical alterations for the maximum (or any) higher prediction probability. We can however recur the replacement procedure multiple times, arriving at questions with larger lexical distance to the original question – see Fig. 6.1 for a schematic overview. For example, in two iterations of named entity replacements, we can alter *"What was Fort Caroline renamed to after the Spanish attack?"* to *"What was Fort Knox renamed to after the Hungarian attack?"*. Similarly, using PoS-consistent perturbations we could in two perturbation steps alter the original question *"Who was the duke in the battle of Hastings?"* to *"Who was the duke in the expedition of Roger?"*

Note that the space of possibilities grows combinatorially with increasing perturbation distance from the original question. Thus at some point it becomes computationally infeasible to comprehensively cover the full space arising from iterated substitutions in an exhaustive search for a successful adversarial attack. To address this, we follow prior work (Feng et al., 2018) that faced a similar problem when examining partial input *deletions* and apply a variation of beam search to narrow the search space. Concretely, during our search we seek to maximise the difference

$$\Delta(x') = P(\hat{y} \mid x') - P(\hat{y} \mid x) \tag{6.3}$$

That is, any $x' \in \mathcal{X}_\mathcal{T}$ for which $\Delta(x') > 0$ is satisfied resembles a successful undersensitivity attack on x. To find such attacks we conduct beam search with a beam of width $b \in \mathbb{N}^+$ up to a maximum perturbation radius $\rho \in \mathbb{N}^+$, corresponding to the maximum search depth. Both these hyperparameters are pre-specified and constrain the computational search budget of the search for an adversarial input. Once a single x' with $\Delta(x') > 0$ has been discovered the search is stopped: an undersensitivity attack has been found. That is, rather than finding the most extreme adversarial attack (with largest value of Δ) we here only aim to find *any* successful undersensitivity attack (with $\Delta > 0$). Using beam search rather than exhaustive search under-estimates the undersensitivity attack rate since the search space is not exhaustively covered. However, the worst-case computational time complexity of this search is linear, rather than the exponential complexity in the worst case of an exhaustive search.

6.2.5 Relation to Attacks in Prior Work

We again emphasise that the type of adversarial attack considered here stands in contrast to other attacks on NLP systems which are based on small, *semantically invariant* input text perturbations (Belinkov and Bisk, 2018; Ebrahimi et al., 2018; Ribeiro et al., 2018b) which highlight oversensitivity problems. Semantic *invariance* comes with stronger requirements and may rely on synonym dictionaries (Ebrahimi et al., 2018) – which are either incomplete or potentially mismatching the context – or paraphrases harvested from back-translation (Iyyer et al., 2018), which can be noisy. Our attack is instead focused on *undersensitivity*, i.e. where the model is stable in its prediction even though it should not be. Consequently the requirements for perturbation spaces that *alter* the question meaning are less difficult to fulfil, and one can rely on sets of named entities and PoS examples automatically extracted from large text collections. In additional contrast to prior attacks (Ebrahimi et al., 2018; Wallace et al., 2019a) we evaluate each perturbed input with a standard forward pass, rather than a first-order Taylor approximation to estimate the output change induced by a change in the input. This is less efficient but exact, and furthermore does not require white-box access to the model and its parameters, as opposed to the attacks formulated by Ebrahimi et al. (2018) and Wallace et al. (2019a).

Our method does not require a human in the loop to adversarially compose questions (Wallace et al., 2019b). On the other hand, it can be seen as connected to previous work (Kang et al., 2018; Minervini and Riedel, 2018) which leverages domain knowledge to generate adversarial inputs. Adopting this perspective, we leverage the idea that modifying, e.g., named entities in a question will likely alter the nature of its information request. The substitution-based approach is furthermore open to extension by restricting the altered expressions under consideration, e.g. based on WordNet (Miller, 1995).

6.2.6 Relation to Prior Work on Undersensitivity

A related viewpoint to undersensitivity is the one of model diagnosis, with the goal of identifying minimal feature sets that are sufficient for a model to form high-confidence predictions (Ribeiro et al., 2018a). In contrast to model diagnosis and other prior work (Feng et al., 2018) showing that it is possible to reduce inputs to minimal input word sequences without changing a model's predictions, we consider concrete natural language alternative questions. We will later furthermore address the observed undersensitivity using additional training data, whereas prior work highlights the problem but does not provide an effective defence (Feng et al., 2018; Ribeiro et al., 2018a). Specifically in dialogue models, undersensitivity has been identified and addressed with a max-margin training approach (Niu and Bansal, 2018). We see our study as an addition and continuation of this previous work on model undersensitivity, with a particular focus set on undersensitivity in extractive RC.

6.2.7 Unanswerable Questions in Reading Comprehension

A central premise of the undersensitivity attack we have described above, is that models have the ability to give a *NoAnswer* prediction. Following the publication of adversarial attacks (Jia and Liang, 2017) on the SQuAD1.1 dataset, the SQuAD2.0 dataset was proposed (Rajpurkar et al., 2018), which includes more than 43,000 new and human-curated unanswerable questions into SQuAD. Another dataset with unanswerable question is NewsQA (Trischler et al., 2017), which comprises questions about news texts. Training on these datasets should result in models that have learned to predict whether questions are answerable or not. We will however see in the subsequent experiments that this ability does not extend to the adversarially chosen unanswerable questions in our undersensitivity attacks. To improve model's performance on unanswerable questions, prior work includes additional verification steps (Hu et al., 2019) or uses synthetic data (Zhu et al., 2019; Alberti et al., 2019), which coincides with one of the adversarial defences we will later test.

6.3 Experiments: Model Vulnerability

6.3.1 Training and Dataset Details

We next conduct experiments using the attacks laid out above to investigate model undersensitivity. We first attack the BERT model (Devlin et al., 2019) fine-tuned on SQuAD2.0 (Rajpurkar et al., 2018), and measure to what extent the model exhibits undersensitivity when adversarially choosing input perturbations. Note that SQuAD2.0 per design contains unanswerable questions in both training and evaluation sets; models are thus trained to predict a *NoAnswer* option where a comprehension question cannot be answered.

In a preliminary pilot experiment, we first train a BERT Large model on the full training set for 2 epochs, where it reaches 78.32%EM and 81.44%F_1, in close range to the results (78.7%EM and 81.9%F_1) reported by Devlin et al. (2019). We then however choose a different training setup as we would like to conduct adversarial attacks on data entirely inaccessible during training: we split off 5% from the original training set for development purposes and retain the remaining 95% for training, stratified by articles. We use this development data to tune hyperparameters and perform early stopping, evaluated every 5,000 steps with a batch size of 16 and maximum patience of 5, and will later tune hyperparameters for defence on it. The original SQUAD2.0 development set is then used as evaluation data, where the model reaches 73.0%EM and 76.5%F_1. We will search for undersensitivity attacks on this entirely held out portion of the dataset.

6.3.2 Attack Details

To compute the perturbation spaces, we collect large sets of string expressions across Named Entity (NE) and PoS types to define the perturbation spaces $\mathcal{T}_E$ and $\mathcal{T}_P$, which we gather from the Wikipedia paragraphs used in the SQUAD2.0 training set, with the pretrained taggers in *spacy*[3] and the Penn Treebank tag set for PoS. This results on average in 5,126 different entities per entity type, and 2,337 different words per PoS type. As the number of possible perturbations to consider for each given question reaches into the thousands, we constrain beam search further: we limit beam search at each iteration step to a maximum of η randomly chosen type-consistent entities, or lexical realisations of PoS tags, re-sampling these selections at each level of the search and for each new attack. Overall the worst case bound on the total computation spent on adversarial search is thus $b \cdot \rho \cdot \eta$ model evaluations per sample (ρ being the perturbation 'radius' or maximum search depth). We set beam width to $b = 5$ throughout. If the number of expressions to potentially be substituted in the question (e.g. the number of named entities) exceeds ρ, the search still explores further perturbations of these expressions different to previously examined perturbations, since η new alternative substitutions are re-sampled randomly at each level of the search.

[3] https://spacy.io

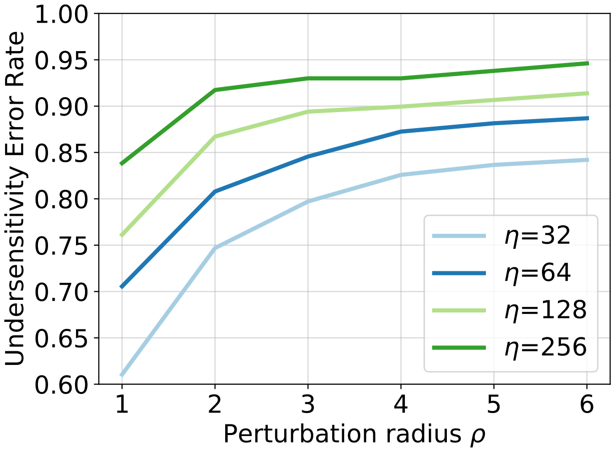


Figure 6.2: BERT LARGE on SQUAD2.0: vulnerability to noisy Part of Speech-perturbation attacks on held out data for differently sized attack spaces (parameter η) and different beam search depth (perturbation radius ρ).

6.3.3 Evaluation: Adversarial Error Rate

Covering the full, exponentially-sized search space defined by iteratively applying transformations to a question would come at significant computational cost, hence the use of beam search as a heuristic. The hyperparameters involved in this beam search do however influence the portion of the perturbation space covered during adversarial search, hence it is important to measure adversarial error rates as a function of the computational budget used. We thus quantify adversarial vulnerability as a function of different computational search budgets, which are defined by different values of η and ρ. We measure vulnerability to the described attack by calculating the proportion of evaluation samples for which at least one undersensitivity attack is found given a computational search budget and name this the *Undersensitivity Error Rate*. We disregard samples where a model predicts *NoAnswer* already for the original question, since altering unanswerable samples likely retains their unanswerability and a successful attack on these does not carry the same implications on undesirable model behaviour.

6.3.4 Experimental Outcome: Model Undersensitivity

Figures 6.2 and 6.3 summarise the undersensitivity error rates for both types of question perturbation across a variety of search budgets. We observe that attacks

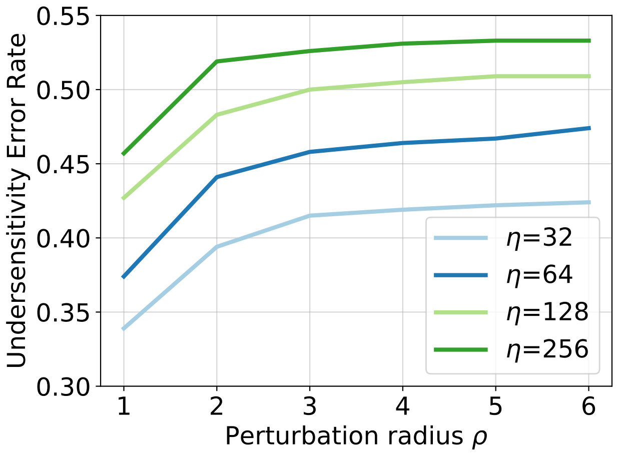


Figure 6.3: BERT LARGE on SQUAD2.0: vulnerability to noisy Named Entity-perturbation attacks on held out data for differently sized attack spaces (parameter η) and different beam search depth (perturbation radius ρ).

based on PoS perturbations can already for very small search budgets ($\eta = 32$, $\rho = 1$) reach more than 60% attack success rates, and this number can be raised to as much as 95% with a larger computational budget. Furthermore we empirically observe a direct and monotonic dependence of undersensitivity error rate on both the maximum perturbation radius ρ, and also on the number η of randomly sampled perturbations at each search level. For perturbations based on NE substitution, we observe a qualitatively very similar dependence on the computational search budget parameters. While for NE-based attacks we observe overall much lower attack success rates than for PoS-based perturbations, we find that more than half of the samples can be successfully attacked when given a sufficient budget. Given the observed progression in adversarial error rates, increasing the attack budgets further (in particular via η) is likely to result in further increased attack rates, although the marginal increases appear to slowly decrease. Note that for samples x where an attack is found, we observe that there often exist not only one x' with $\Delta(x') > 0$, but several successful attacks.

In summary, these results show that the BERT model, trained on SQUAD2.0, is vulnerable to undersensitivity attacks: a considerable number of samples can successfully be attacked for both perturbation types tested. This suggests that the model prediction procedure indeed lacks specificity towards particular aspects of the infor-

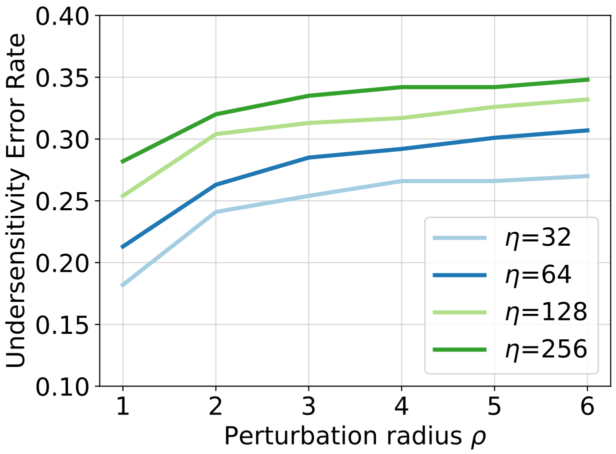


Figure 6.4: Vulnerability to undersensitivity attacks on NEWSQA.

mation request formulated in the comprehension questions. Even though trained to tell when questions are unanswerable, the model can be brought to fail more than half of the time when facing adversarially selected unanswerable questions, already with a limited search budget.

6.3.5 Experiment: NEWSQA

To test undersensitivity on a second dataset, we next consider the NEWSQA dataset (Trischler et al., 2017), which – like SQUAD2.0– contains unanswerable questions. As annotators often do not fully agree on their annotation in NEWSQA, we opt for a conservative choice and filter the dataset, retaining only those samples with the same majority annotation, following the pre-processing pipeline of prior work (Talmor and Berant, 2019).

Fig. 6.4 depicts the vulnerability of a BERT LARGE model trained and evaluated on NEWSQA under attacks using NE perturbations. We again observe a considerable proportion of undersensitivity errors – albeit lower rates than for SQUAD2.0; the overall dependence of the undersensitivity error rate on the computational attack budget parameters is again similar to the one previously observed. The presence of undersensitivity errors also on NEWSQA suggests that this phenomenon is not confined to one particular dataset, but a more general issue of RC models – even when trained on datasets including unanswerable samples.

6.3.6 Side Experiment: SQuAD1.1

We next briefly investigate undersensitivity attacks using NE-based perturbations on SQuAD1.1. This allows for a tentative assessment of the addition of unanswerable samples into the training data in the context of undersensitivity. Since there is however no *NoAnswer* label in SQuAD1.1, the appropriate behaviour for unanswerable questions is not well-defined, complicating the interpretation of results regarding undersensitivity.

Using exactly the same model and experimental setup as for SQuAD2.0 (but trained on SQuAD1.1 and without the ability to predict *NoAnswer*), we see that when BERT is trained on SQuAD1.1, it is even more adversarially vulnerable (in the sense of a large sample fraction for which alternative questions can be found that satisfy Inequality 6.2). The "undersensitivity error rate"[4] reaches 70% already with a search budget of $\eta = 32$; $\rho = 1$ (compared to 34% on SQuAD2.0). That is, undersensitivity is a much larger issue for SQuAD1.1 than for SQuAD2.0. The notable drop in undersensitivity error rate between versions 1.1 and 2.0 of the SQuAD dataset suggests that adding unanswerable questions might be an effective means to mitigate model undersensitivity – a route we will pursue later in Section 6.6.

6.4 Qualitative Analysis of Attacks

The perturbation spaces for the undersensitivity attack are designed to retain the syntactic or semantic structure of the original question, while altering its concrete information request. However the PoS and NER tag predictions contain potential errors, and the introduced substitutions are not guaranteed to always result in syntactically and semantically coherent expressions, or indeed have a different correct answer than the original question.[5] To gauge the extent of this potential issue, we

[4] Referring to this metric as an "error rate" is not necessarily justified given the inability of SQuAD1.1 models to predict a *NoAnswer* label.

[5] There are different opinions on whether or not to consider successful adversarial attacks based on ill-formed questions as problematic. On the one hand, a model's desirable behaviour is not well defined for such inputs; evaluating a model on non-sensical comprehension questions is in itself not very useful. On the other hand it can – like for well-formed questions – demonstrate that the question processing of the model is not as sensitive as it perhaps should be to the expression that has been substituted from the original question, even if its replacement is nonsensical.

	PoS	**NE**
Syntax Error	10%	6%
Semantically Incoherent	24%	5%
Same Answer	15%	5%
Valid Attack	51%	84%

Table 6.2: Quantitative results for the analysis of undersensitivity attack samples for both PoS and named entity (NE) perturbations.

inspect 100 successful attacks conducted at $\rho = 6$ and $\eta = 256$ on SQuAD2.0, both for PoS-based and NE-based perturbations. We label them according to the following schema:

1. **Syntax Error:** e.g. *"What would platform lower if there were fewer people?"*. Such cases are mostly due to cascading errors stemming from wrong NE / PoS tag predictions.

2. **Semantically Incoherent:** e.g. *"Who built the monks?"*

3. **Same Answer:** Questions that require the same correct answer as the original, e.g. due to a paraphrase.

4. **Valid Attack:** Questions that would either demand a different answer than the original question or are unanswerable, given the text (*e.g. "When did the United States withdraw from the Bretton Woods Accord?"* and its perturbed version *"When did Tuvalu withdraw from the Bretton Woods Accord?"*)

When labelling the attacks according to this schema, a single option of these four alternatives is selected. Where several could apply (e.g. with both a syntactic *and* a semantic incoherence issue) labels are selected in the above presented order with higher items receiving higher priority. For illustration, Table 6.3 shows several example attacks alongside their selected labels.

In Table 6.2 we then summarise the results of this analysis quantitatively for both perturbation types. First, we observe that a non-negligible portion of altered

Original / Modified Question	**Prediction**	**Annotation**	**Scores**
What city in Victoria is called the cricket ground of Australia the Delhi Metro Rail Corporation Limited?	Melbourne	valid	0.63/0.75
What ethnic neighborhood in Fresno Kilbride had primarily Japanese residents in 1940?	Chinatown	valid	0.998/0.999
What are some of the accepted general principles of European Union Al-Andalus law?	fundamental rights [...]	valid	0.59/0.61
The Mitchell Tower MIT is designed to look like what Oxford tower?	Magdalen Tower	valid	0.96/0.97
What were the annual every year carriage fees for the channels?	£30m	same answer	0.95/0.97
What percentage of Victorians are Christian Girlish?	61.1%	valid	0.92/0.93
What does the EU's legitimacy digimon rest on?	the ultimate authority of [...]	valid	0.38/0.40
What is Jacksonville's hottest recorded temperature atm?	104°F	valid	0.60/0.62
Which plateau is the left part achievement of Warsaw on?	moraine	semantic inconsistency	0.52/0.58
Q: Who leads the Student commissioning Government? **Paragraph**: [...] Student Government is made up of graduate and undergraduate students [...]. It is led by an Executive Committee, chaired by [...]	an Executive Committee	same answer	0.61/0.65

Table 6.3: Example adversarial questions (original, attack), together with their annotation as either a valid counterexample or other type. Top 5 rows: perturbations based on Named Entity substitutions. Bottom 5 rows: perturbations based on PoS substitutions. The last column lists the prediction probability assigned by the model for the original / attack question. For clarification of the last example the respective WIKIPEDIA paragraph is shown.

questions indeed has some form of syntax error or incoherent semantics, in particular for PoS-based perturbations. About a quarter of the attacks based on PoS-perturbations have some sort of semantic incoherence (e.g. *"What year did the case go before the supreme court?"* vs. its perturbed version *"What scorer did the case go before the supreme court?"*).

Questions with the same correct answer are comparatively rare, though again more prevalent with PoS-based perturbations than with the NE-based ones. Overall, with 84% the attacks based on NE perturbations are more often valid, featuring fewer problems of any type compared to those based on PoS perturbations, for which only about half of all attacks are valid.

6.4.1 Discussion

Our analysis shows that far from all of the identified attacks are indeed valid ones. For example, with approximately 51% of valid PoS attacks and a vulnerability of 95% (η=256, $\rho = 6$), only about half of the attacks can be considered valid ($0.51 \cdot 0.95 \approx 0.48$). But even with our restriction of using beam search and considering only η perturbations per level, there is usually more than one successful attack per sample. Our qualitative analysis, on the other hand, only considers one of them. The above calculated number of 48% is hence rather a lower bound on the extent of the model's adversarial vulnerability to valid and well-formed questions.

It is further worth noting that the observed problems, especially for PoS-type perturbations, are a consequence of imperfectly characterising the natural language perturbation space to search in: not all perturbations result in well-formed questions, thus introducing noise into the adversarial search space. While the results of the qualitative analysis diminish the empirical weight of the quantitative results of the undersensitivity experiment, we believe that the underlying issue is one of an imperfect metric derived using noisy attack spaces (thus also counting invalid attacks), rather than a lack of model undersensitivity to valid perturbations. Nevertheless, since the named entity-based attacks have a substantially larger fraction of valid alternative questions as attack, we will for the remainder of this chapter focus our study on NE-based attacks.

6.5 Characterising Vulnerable Data Points

Even with noisy attacks, we have empirically established that the BERT model is vulnerable to undersensitivity adversaries. However not all samples were successfully attacked. Naturally, this gives rise to the following question: what distinguishes those samples that can from those samples that cannot be attacked by an undersensitivity adversary? We will next investigate various characteristics of the questions and perturbations, with the aim of understanding causes of the undersensitivity vulnerability. For all subsequent characterisations we rely on adversarial attacks on SQuAD2.0 based on NE perturbations computed with an adversarial search budget of $\rho = 6$ and $\eta = 256$.

A first observation is that questions that can be attacked tend to produce lower original prediction probabilities with an average of only 72.9% prediction confidence, compared to 83.8% for questions without a successful attack found using the given search budget. That is, there exists a direct inverse link between a model's original prediction probability and a sample's vulnerability to an undersensitivity attack. The adversarially chosen questions then have an average probability of 78.2%, i.e. a notable gap to the original questions.[6]

Second, the BERT model generally struggles to give the correct prediction for the vulnerable samples: they are less likely to be given the correct answer prediction by the model. Concretely, evaluation metrics for vulnerable examples are only 56.4% / 69.6% EM/F_1, compared to 73.0%/76.5% on the full dataset, or 77.1%/83.3% on answerable questions.[7] An RC model's vulnerability to an undersensitivity attack on a particular sample is thus a negative predictor for the model's accuracy on this sample. A possible explanation for this observation is that if a model builds a poor representation of the particular information request specified in the comprehension question, which is reflected in its vulnerability to an undersensitivity attack, then it is also more likely to be wrong about its prediction.

Next, we observe that questions of vulnerable samples are on average slightly longer: their mean length is 12.3 tokens, compared to 11.1 tokens for questions not found vulnerable to the undersensitivity attack. When we consider the distribution of different question types (*What, Who, When, ...*) for both successfully attacked and samples without attack, we do not observe notable differences apart from the single most frequent question type *What*; it is more prevalent among samples without attack (56.4%) than under successfully attacked samples (42.1%). This is by far the most common question type; it is furthermore comparatively open-ended, i.e. does not prescribe particular type constraints to its answer, as e.g., a *Where* question would require a location, or a *When* question usually requires a date. A possible

[6] It is worth noting again that the adversarial search in our above experiment halts as soon as a single question with higher answer probability than the original is found; hence it is likely that a more thorough search can identify adversarial attacks with higher probabilities, on average.

[7] Answerable questions are those where a is not the *NoAnswer* label.

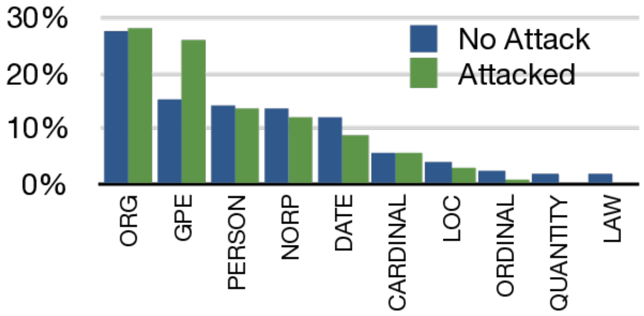


Figure 6.5: Named entity type characteristics of successfully vs. unsuccessfully attacked samples.

explanation for the prevalence of the *What* questions among the samples without attack is that when the model forms its predictions on these questions, it cannot rely as much on type constraints to the answer as predictive cue and shortcut, and is thus less prone to adversarial exploitation. In Section 6.7.3 we will pursue this avenue further and examine undersensitivity behaviour in the context of very predictive type constraints.

Finally, in Fig. 6.5 a histogram shows the 10 most common named entity tags appearing in samples without attack, contrasting them to the corresponding fraction in successfully attacked samples. Besides one exception, the distributions are remarkably similar: undersensitivity can be found across entity types. Questions with geopolitical entities (*GPE*), however, are particularly error-prone. A possible explanation for this observation could be that (non-contextualised) embeddings tend to cluster geopolitical entities (e.g. countries), thus rendering it harder for a model to distinguish them based on their respective embeddings (Mikolov et al., 2013b). Such vector similarity could be the reason for the model's lack of discrimination between these entities, although the GPEs here are multi-word expressions and their embeddings computed based on further contextual information (Devlin et al., 2019).

6.5.1 Transferability of Adversarial Attacks between Models

Prior work has pointed out that the same adversarial attacks can generalise *between* different models (Feng et al., 2018; Wallace et al., 2019a), which would suggest that adversarial vulnerability might be a consequence of the training data, the training method, or the pre-training data – rather than the particular model architecture. We conduct an experiment where we train the RoBERTa model (Liu et al., 2019), also on SQuAD2.0, and carry out NE-based undersensitivity attacks with $\rho = 6$ and $\eta = 256$.

With 35%, the attack rates for RoBERTa are lower than for BERT (53%), nevertheless a considerable number of samples is vulnerable. That is, the improved pre-training of RoBERTa, which leads to improved nominal generalisation results on the SQuAD2.0 test set compared to BERT, also alleviates model undersensitivity, which for BERT we previously observed is negatively correlated with nominal generalisation performance.

But indeed, the more striking result is about the transferability of adversarial vulnerability: there is a substantial overlap between the sets of vulnerable samples of the two models. While overall successfully attacked less often, when considering only those samples for which RoBERTa was successfully attacked, BERT has an undersensitivity error rate of 90.7%. That is, the vulnerability of a sample to an undersensitivity attack on RoBERTa predicts the vulnerability of the same sample also for BERT.

Moreover, even concrete adversarial inputs chosen based on RoBERTa can transfer to the BERT model. Feeding specific pairs of original and adversarial question selected using the RoBERTA model as inputs to the BERT model leads to a successful attack on BERT for 17.5% of these samples. For illustration, one such example of an attack that is transferable between the models is shown in Table 6.4.

The large overlap of vulnerable data points between the two models suggests that the vulnerability to undersensitivity attacks might partially be dependent on the training data used to tune these models. In the next section we will consider altering the data points that the model is exposed to during training, and observe that changing the training data indeed affects the model's undersensitivity behaviour.

Given Text	James Hutton is often viewed as the first modern geologist. In 1785 he presented a paper entitled Theory of the Earth to the Royal Society of Edinburgh. [...]	
Question:	In 1785 James Hutton Jacob Ettlinger presented what paper to the Royal Society of Edinburgh?	
Prediction (BERT):	Theory of the Earth 99.9%/99.9%	(Difference: +0.05%)
Prediction (RoBERTA):	Theory of the Earth 99.3%/99.6%	(Difference: +0.30%)

Table 6.4: An example where a concrete adversarial undersensitivity attack identified on RoBERTA is transferable to a BERT model, where it also presents a valid attack. Expressions highlighted in green/red belong to the original/adversarial question.

6.6 Defending Against Undersensitivity Attacks

6.6.1 Defence Baselines

We will now investigate methods for mitigating a model's undersensitivity. Prior work has considered both data augmentation and adversarial training as methods to achieve more robust models with decreased adversarial vulnerability; we will conduct experiments with both. However, introducing an additional objective to improve a model's robustness can negatively impact standard test metrics, and there is a natural trade-off between performance on one particular test set, and performance on a dataset comprising adversarial inputs (Tsipras et al., 2019).

The two defences we test both make use of additional training samples in an attempt to improve a model's undersensitivity. We denote with Ω the standard (original) training dataset; our defence methods then each include a new set of (sampled) data points Ω' that contributes to the training loss. Concretely, we perform both data augmentation and adversarial training by adding a corresponding loss term to the standard log-likelihood training objective:

$$\mathcal{L}^{Total} = \mathcal{L}^{llh}(\Omega) + \lambda \cdot \mathcal{L}^{llh}(\Omega') \tag{6.4}$$

where Ω is the standard training data, fit with a discriminative log-likelihood objective, Ω' either a set of (sampled) augmentation data points, or of successful adversarial attacks where they exist, and $\lambda > 0$ a hyperparameter. In data augmentation, we randomly sample perturbed input questions, whereas in adversarial training we perform an adversarial search to identify them. In both cases, alternative data points in Ω' will be fit using a log-likelihood objective to a *NULL* label corresponding to the *NoAnswer* prediction on the SQUAD2.0 and NEWSQA datasets we examine.

In data augmentation we sample randomly perturbed input questions from the same perturbation spaces used for the undersensitivity attacks. We update these data points throughout training, and we sample them in equal proportion to the original data, but weigh their relative contribution to the loss using λ. Irrespective of the probability assigned for the answer to the original question, the target label for the new data points in Ω' is the *NULL* label. During training, the model is then not only exposed to the original training samples, but also to structurally similar but unanswerable samples. Our hope is that the presence of these closely related 'negative' samples encourages the model to become more specific to the given entities in the question, and thus less undersensitive.

The second defence we test is adversarial training: here we perform, in every training step, an adversarial search to identify successful adversarial attacks. As this requires several forward passes in the inner training loop, we restrict ourselves to a relatively low adversarial search budget ($\eta = 32, \rho = 1$). We emphasise that while the original training data Ω remains constant throughout training, we continuously update Ω' in order to reflect adversarial samples based on the current model.

6.6.2 Experimental Setup

We train a BERT LARGE model on SQUAD2.0 and tune the hyperparameter $\lambda \in \{0.0, 0.01, 0.1, 0.25, 0.5, 0.75, 1.0, 2.0\}$. We tune the threshold for predicting *NoAnswer* based on validation data and report results on our test set (the original SQUAD2.0 *dev* set). All experiments are run with batch size 16 and NE perturbations for both attacks and defences. Where no attack is found for a given question during adversarial training, we redraw standard samples from the original training data and use these instead of adversarial samples (with their original, annotated label). We evaluate the model on its validation data every 5,000 training steps and perform early stopping with a patience of 5 based on the F_1 score on the validation data. Following the same experimental protocol, we also experiment with a BERT

SQuAD2.0	**Undersensitivity Error Rate**				**HasAns**		**NoAns**	**Overall**	
Adv. budget η	@32	@64	@128	@256	EM	F_1	EM/F1	EM	F_1
Bert Large	44.0	50.3	52.7	54.7	**70.1**	**77.1**	76.0	73.0	76.5
+ Data Augment.	**4.5**	**9.1**	**11.9**	**18.9**	66.1	72.2	**80.7**	**73.4**	76.5
+ Adv. Training	11.0	15.9	22.8	28.3	69.0	76.4	77.1	73.0	**76.7**

Table 6.5: Breakdown of undersensitivity error rates (lower is better) for different adversarial search budgets η, and standard performance metrics (EM, F_1; higher is better) on different subsets of the SQuAD2.0 evaluation data, all in [%].

NewsQA	**Undersensitivity Error Rate**				**HasAns**		**NoAns**	**Overall**	
Adv. budget η	@32	@64	@128	@256	EM	F_1	EM/F1	EM	F_1
Bert Base	34.2	34.7	36.4	37.3	**41.6**	53.1	61.6	45.7	54.8
+ Data Augment.	**7.1**	**11.6**	**17.5**	**20.8**	41.5	**53.6**	62.1	**45.8**	**55.3**
+ Adv. Training	20.1	24.1	26.9	29.1	39.0	50.4	**67.1**	44.8	53.9

Table 6.6: Breakdown of undersensitivity error rates (lower is better) for different adversarial search budgets η, and standard performance metrics (EM, F_1; higher is better) on different subsets of the NewsQA evaluation data, all in [%].

Base model on the NewsQA dataset.

6.6.3 Results and Discussion

Can the two defence strategies reduce a model's undersensitivity error rate? How are the standard performance metrics affected by this? The results of the experiments for the two datasets can be found in Table 6.5 and Table 6.6.

A first, and very robust observation is that both data augmentation and adversarial training substantially reduce the number of undersensitivity errors the model commits – consistently across all adversarial search budgets, and consistently across both datasets. That is, the addition of "negative", unanswerable training samples can indeed reduce a model's undersensitivity. Both defence methods are effective and relieve – but do not entirely rid – the model of undersensitivity errors. Notably the improved robustness – especially for data augmentation – is possible without sacrificing performance in the EM and F_1 standard metrics; we even see slight improvements.

Next we observe that data augmentation is generally a more effective defence training strategy than adversarial training (at least in the form tested here). This potentially hints at adversarial overfitting, but might also be a consequence of the relatively low adversarial search budget used during training; further experiments would be needed to draw more robust conclusions.

Finally, a closer inspection of *how* performance changes on answerable (*HasAns*) vs. unanswerable (*NoAns*) samples of the datasets reveals that models with modified training objectives have improved performance on unanswerable samples, while sacrificing performance on answerable samples.[8] This suggests that the trained models – even though similar in standard metrics – evolve on different paths during training, and that modifying their training objective prioritises fitting unanswerable questions to a higher degree.

6.7 Positive Consequences of Reduced Undersensitivity

6.7.1 Generalisation to New Perturbations

The previously reported results in Tables 6.5 and 6.6 are computed using the same perturbations at training and evaluation time. The perturbation space is relatively large, with on average several thousand entries per NE tag, and evaluation questions are furthermore posed about a set of articles disjoint from those used during training. Nevertheless there is a potential risk of overfitting to the particular perturbations used in the adversarial defences. Does the improved undersensitivity behaviour for the set of perturbations used during training translate to new perturbations with previously unseen named entities?

To measure the extent to which the defences generalise also to new, held out sets of perturbations, we assemble a new, disjoint perturbation space, and evaluate models on attacks with respect to these perturbations. Named entities are chosen from English WIKIPEDIA using the same method as for the training perturbation spaces, and chosen such that they are disjoint from the training perturbation space. Furthermore we ensure that the new space has an identical number of NE expressions per NE type as before, to rule out that differences in size are responsible for

[8] Note that the *NoAnswer* prediction threshold is fine-tuned after training on the respective validation sets.

SQuAD2.0	**Undersensitivity Error Rate**			
Adv. budget η	@32	@64	@128	@256
BERT LARGE	40.7	45.2	48.6	51.7
+ Data Augment.	**4.8**	**7.9**	**11.9**	**20.7**
+ Adv. Training	9.2	12.2	16.5	23.8

Table 6.7: Breakdown of undersensitivity error rate on SQuAD2.0 with a held-out perturbation space (lower is better).

NEWSQA	**Undersensitivity Error Rate**			
Adv. budget η	@32	@64	@128	@256
BERT BASE	32.8	33.9	35.0	36.2
+ Data Augment.	**3.9**	**6.5**	**11.9**	**17.5**
+ Adv. Training	17.6	20.7	25.4	28.5

Table 6.8: Breakdown of undersensitivity error rate on NEWSQA with a held-out perturbation space (lower is better).

any of our observations. We then conduct adversarial attacks using these new attack spaces on the previously trained models. Vulnerability results for these new, held-out perturbation spaces, disjoint from those used during training, can be found in Table 6.7 for SQuAD2.0, and in Table 6.8 for NEWSQA.

We observe that both the vulnerability rates of the original model, and – remarkably – also the relative and absolute success of the defences transfers to the new set of perturbations. This demonstrates that adding closely related unanswerable samples during training leads to benefits that are not confined to a narrow set of specific perturbations used during training, but hold more broadly for a similarly large set of new perturbations as well.

6.7.2 Improved Robustness on Adversarial SQuAD

How does data augmentation with closely related unanswerable samples affect the model's robustness to adversarial samples from an *oversensitivity* attack? We next compare the original BERT LARGE model, and a BERT LARGE model trained with the data augmentation defence on the ADDSENT and ADDONESENT datasets (Jia and Liang, 2017).

These datasets contain adversarially composed samples about WIKIPEDIA ar-

	ADDSENT		ADDONESENT		DEV 2.0	
	EM	**F_1**	**EM**	**F_1**	**EM**	**F_1**
BERT Large	61.3	66.0	70.1	74.9	78.3	81.4
BERT Large+Aug.	**64.0**	**70.3**	**70.2**	**76.5**	**78.9**	**82.1**

Table 6.9: Comparison between BERT LARGE and BERT LARGE + Data Augmentation Training on two sets of adversarial examples: ADDSENT and ADDONESENT from Jia and Liang (2017).

ticles and test in particular how well models react to the adversarial injection of new sentences (with high degrees of lexical overlap to the question) into the given paragraph. It has been shown that across a wide range of models, RC model performance on these questions drops compared to standard SQUAD questions (Jia and Liang, 2017), demonstrating that models overly rely on surface cues – in particular lexical overlap and answer type – when forming their prediction. We train the BERT LARGE model on the full SQUAD2.0 training set; for data augmentation we set λ to 1.0.

Our results, summarised in Table 6.9, show that the data augmentation defence improves the EM and F_1 scores of BERT LARGE on ADDSENT and ADDONESENT, boosting F_1 by 4.3 and 1.6 points on the two datasets, respectively. That is, when adding structurally related "negative" examples during training, the model's robustness when adversarially inserting distracting text into the given paragraph improves. This demonstrates that adding such training samples positively affects the model's specificity not only when processing the information request formulated in the question, but also in the given paragraph – it does not get distracted as easily.

6.7.3 Generalisation in a Biased Data Setting

As we have already observed several times in this thesis, datasets for high-level NLP tasks often come with annotation and selection biases. Models can then learn to exploit shortcut triggers which are dataset- but not task-specific (Jia and Liang, 2017; Gururangan et al., 2018). For example, a model might be confronted with question / paragraph pairs which only ever contain one type-consistent answer span, e.g. mention only a single number in the paragraph for a "*How many...?*" question.

	Person		**Date**		**Numerical**	
	EM	F_1	EM	F_1	EM	F_1
Data Setting I (w/ data bias): GQA	53.1	61.9	64.7	72.5	**58.5**	**67.6**
BERT BASE	66.0	72.5	67.1	72.0	46.6	54.5
BERT BASE + Augmentation	**67.4**	**72.8**	**68.1**	**74.4**	56.3	64.5
Data Setting II (w/ data bias): BERT BASE	55.9	63.1	48.9	58.2	38.7	48.0
BERT BASE + Augmentation	**59.1**	**66.6**	**58.4**	**65.6**	**48.7**	**58.9**
Data Setting III (w/o data bias): BERT BASE	69.2	78.1	73.2	81.7	69.6	80.5

Table 6.10: Training with augmented data leads to improved generalisation under train/test distribution mismatch (w/ data bias, Data Settings I and II). Data Setting I: direct comparison to prior work (Lewis and Fan, 2019). Data Setting II: Dataset splits stratified by article + held out development set; Data Setting III: Control experiment shuffling all data points, i.e. with no train/test distribution mismatch.

To solve the task it is then sufficient – and indeed a viable strategy – for the model to learn to pick out numbers from text – irrespective of other information given in the question. Such a model might then have trouble generalising its RC capabilities to articles that mention several numbers, as it has never learned that it is necessary to take into account other relevant information specified in the comprehension question.

We now test models in such a scenario: a model is trained on SQUAD1.1 questions with paragraphs containing only a single type-consistent answer expression for either a person, date, or numerical answer. At test time, we present it with question / article pairs of the same respective question types, but now there are *multiple* possible type-consistent answers in the paragraph. We obtain such data after correspondence from the authors of prior work (Lewis and Fan, 2019), who first described this biased data scenario. We will test three data settings (I-III) for three categories of answer types (*Person, Date, Numerical*). The results of all these experiments are summarised in Table 6.10.

In the first setting (Data Setting I), we compare both a standard BERT BASE transformer model, and a model trained to be less vulnerable to undersensitivity attacks using the data augmentation defence. We follow the data setup used in prior work (Lewis and Fan, 2019) for direct comparison to the GQA model (not holding aside a dedicated validation set, selecting hyperparameters based on the final evaluation set performance). We find that the augmentation defence improves the model's performance on the test set with a different data bias, for all three data categories (*Person*, *Date*, *Numerical*). Furthermore, the model outperforms GQA (Lewis and Fan, 2019) in two of the three subtasks, although this may be partially due to the usage of BERT.

While allowing for a direct comparison with prior work, the hyperparameter selection in Data Setting I is not ideal. Thus, in Data Setting II we conduct the same experiment once again, splitting the dataset into a training, development, and a test set. We train on the same training data as before, but split the original test set with a 40/60% split, stratified by article, into development and test data.[9] Again, we observe considerable improvements when using data augmentation – across metrics and all three types of answer categories.

Finally, in Data Setting III we perform a control experiment: we join and shuffle *all* data points from train / dev / test (of each question type, respectively), and split the dataset into new parts of the same sizes as before in Data Setting II. The resulting data distribution is thus identical during training and evaluation, i.e. we observe both one *and* several type-consistent answers during training and during evaluation (w/o data bias setting). We observe a much stronger generalisation model performance in Data Setting III, which does not have the data bias of Settings I and II. The considerable gap to the previously computed results confirms that models trained on biased data with only one type-consistent answer (Settings I and II) indeed learn to rely on the shallow cue of type consistency for predicting the answer – which negatively impacts the model's performance on the evaluation questions with several type-consistent answers in Settings I and II. Training the model with structurally similar unanswerable samples based on NE perturbations – as is done in data augmentation – considerably reduces this gap. This indicates that the altered training data encourages the model to rely on information other than merely type consistency; consequently it generalises better to the evaluation questions in Settings I and II.

[9] Note that 40% and 60% are only approximate due to stratification by article; the closest possible split point is selected.

These experiments demonstrate that the negative training signal stemming from related – but unanswerable – questions counterbalances the signal from answerable questions in such a way that the model learns to better take into account the relevant information in the question. This allows it to correctly distinguish among several type-consistent answer possibilities in the text, which the standard BERT BASE model does not learn well.

6.8 Conclusion

In this chapter we have investigated a problematic behaviour of RC models – being undersensitive, i.e. overly stable in their predictions when given semantically altered questions. Undersensitivity focuses on a model's excessive prediction invariance, i.e. the problem that when input text is meaningfully changed, the model's prediction does not change even though it should. This makes it a complementary problem to most prior work on adversarial attacks in NLP, which has studied semantically invariant text perturbations that cause a model's prediction to change when it should not. We have analysed the resulting attacks qualitatively, established their level of noisiness, and identified characteristics that distinguish successfully attacked samples from those without an attack. Furthermore, the prevalence of successfully attacked samples changes depending on the availability of structurally similar, but unanswerable questions during model training. We summarise the answers to our initially posed research questions as follows:

1. **How can RC model undersensitivity to changes in the question be evaluated using natural language inputs?** We have formulated an adversarial attack which searches among semantic variations of comprehension questions for which a model still erroneously produces the same answer as the original question – and with an even higher probability. We have considered both PoS and NE perturbations as the basis for semantic variations in the input question, and found that the latter produces a higher rate of well-formed questions and valid attacks than the former.

2. **To what extent is the commonly used BERT model undersensitive to ad-**

versarially chosen input perturbations? Despite comprising unanswerable questions, fine-tuning BERT on SQuAD2.0 and NewsQA leaves it vulnerable to the undersensitivity attack: it commits a substantial fraction of errors on noisy adversarially generated questions: at least 95% for PoS perturbations, and at least 54% for NE perturbations on SQuAD2.0.

3. **How can adversarially vulnerable samples be characterised and distinguished?** Vulnerable samples have, on average, lower prediction probabilities, lower accuracy, fewer *What* questions, and more questions mentioning geopolitical entities (GPE) than samples for which the adversarial attack is unsuccessful.

4. **Can the two adversarial defence strategies of data augmentation and adversarial training help alleviate the undersensitivity problem?** Both methods are effective and substantially reduce the model's undersensitivity error rates. This holds true both for perturbations seen during training, as well as new, unseen perturbation spaces. Overall, data augmentation is more effective than adversarial training.

5. **How does training models to be more sensitive to input changes affect their behaviour?** The experiments on adversarial defences have demonstrated that the addition of closely related, but unanswerable questions to the model's training data has multiple benefits: not only does it not sacrifice standard performance, but it does leave the model more robust towards adversarial exploitation (both to undersensitivity attacks and an oversensitivity attack), and less prone to reliance on shallow type consistency heuristics. These are desirable properties of an RC model, yet not directly reflected in standard evaluation metrics.

The fact of a model's poor performance on adversarially selected inputs highlights model shortcomings which nominal test set EM and F_1 do not measure. Since neural RC models are potent data approximation tools, they are prone to learning predictive shortcuts that help them predict the correct answer, without necessarily taking into account all relevant information in the input. As the space of RC input data is large and complex, a sufficient amount of closely related counterexamples may not naturally appear in a crowd-sourced dataset like SQUAD. The discriminative learning setup then leads to model behaviours that can be at odds with our notion of text understanding, and dedicated adversaries can exploit this.

By adding closely related synthetic unanswerable samples to the training data, we showed that we can direct the model towards learning associations that better take into account the entity information in the information request formulated in a comprehension question, rather than learning shortcuts and relying on shallow predictive cues.

More generally, RC evaluation metrics (and the data points used to derive these metrics) should reflect what we would like a model to learn; we see a model's undersensitivity error rate as one such facet that can help us evaluate a model's behaviour and level of reading comprehension.

Chapter 7

Formally Verifying an Undersensitivity Specification

This chapter is based on previously published work (Welbl et al., 2020a). The mathematical derivations and experiments were conceived and conducted by the main author, in close dialogue with the co-authors. IBP was implemented by collaborators, its particular adaptation to the DAM model was implemented by the thesis author.

7.1 Introduction

7.1.1 Adversarial Search in an Exponentially Large Space

We have in the previous chapter explored a model's vulnerability to adversarially chosen input changes that highlight undersensitivity. Two defences that we have considered against it – data augmentation, as well as adversarial training – are both effective at reducing the extent of the issue, but do not resolve it entirely. A substantial number of examples can still be attacked even after applying the defences: for example, the BERT LARGE model on SQUAD2.0, trained with data augmentation and adversarial training still has undersensitivity error rates of 18.9% and 28.3%, respectively (ρ=256; cf. Table 6.5; cf. Table 6.6 for similar observations regarding NEWSQA). We also saw that undersensitivity error rates crucially depend on the computational budget allocated for the adversarial search: increasing the search budget also increases the error rate. Prior work has established that either using stronger adversarial attacks or a less restrictive search heuristic can identify new adversarial inputs (Carlini and Wagner, 2017; Athalye et al., 2018; Uesato et al., 2018). It is thus unclear whether the previously observed improvements in undersensitivity error rate actually eliminate the possibility of such adversarial attacks, or whether they merely raise the associated cost of finding them.

In this chapter we will explore a more rigorous measure against model undersensitivity: the formal *verification* of a specification against undersensitivity. We will formulate this specification for a neural model with the aim of ruling out the possibility that *any* undersensitivity attack can be found – using an unlimited computational search budget. Naturally, a central issue with comprehensively identifying all adversarial attacks (or ruling out their existence) is the exponentially large perturbation space of the undersensitivity attack, which arises from deleting or replacing arbitrary subsets of a given input sequence. The associated search is costly; ruling out the presence of any adversarial input requires covering the space in its entirety.

Prior work on undersensitivity thus relies on heuristics, such as beam search (Feng et al., 2018) – which we also relied on in the previous Chapter 6 –, or bandits (Ribeiro et al., 2018a). These heuristics are useful to *identify* successful attacks, but do not exhaustively cover the full perturbation space. Consequently they cannot provide a guarantee on the non-existence of adversarial inputs (e.g. after applying defences) – they might just fail to identify them. Performing an exhaustive search, on the other hand, can rule out the presence of any adversarial sample, but the computational cost associated with it renders this approach impractical due to the exponential growth of the search space with the length of the input text.

7.1.2 Verification Using Interval Bound Propagation

To overcome the problem of exponential computational cost associated with performing exhaustive search, we will make use of Interval Bound Propagation (IBP; Gowal et al. (2018); Mirman et al. (2018)): a formal verification method to verify specifications in neural network models. Compared to adversarial search, verification begins from a complementary viewpoint and methodological angle: it aims to provide mathematically provable certificates that a given specification on model behaviour is verifiably fulfilled.

As a verification method, IBP is very efficient: it can cover an exponentially large perturbation space with constant time and memory overhead. In this chapter we will use IBP to comprehensively cover the full perturbation space associated with undersensitivity attacks in its entirety, and thus determine whether any successful undersensitivity attack can be found. This will be achieved by first formulating, and then using IBP to verify a specification about the undersensitivity of a model. IBP is a versatile methodology: it is both useful for evaluation purposes, to verify if a given model specification is satisfied; it is however also useful during model training: an auxiliary loss can be derived to *direct* models towards verifiably adhering to the given specification.

7.1.3 Task Simplifications

The formal verification of specifications against adversarial attacks on neural NLP models is a research direction that has only very recently received attention in the research community (Huang et al., 2019; Jia et al., 2019). It still suffers from a number of methodological shortcomings, which will be discussed throughout this chapter. Rather than providing a definite solution, we aim at exploring to what extent IBP-based verification can be useful in a simplified setting to address undersensitivity, and observe basic challenges associated with the use of this methodology. Compared to the previous chapter, we modify the setting of our investigation along three axes.

First, rather than investigating extractive RC, we will focus on the related natural language understanding problem of Natural Language Inference (NLI; Dagan et al. (2006); Bowman et al. (2015)). The NLI task consists in determining whether a given premise entails a given hypothesis – both natural language statements. The development of sufficiently large data resources for NLI (Bowman et al., 2015; Williams et al., 2018) has led to a flourishing field of neural modelling (Rocktäschel et al., 2016; Parikh et al., 2016; Chen et al., 2017b) inter alia. Prior work on adversarial inputs in NLI includes adversarially chosen logical inconsistencies (Minervini and Riedel, 2018), adversaries relying on background knowledge (Kang et al., 2018) and lexical entailment relationships (Glockner et al., 2018). But NLI was also among the tasks for which undersensitivity was first established (Feng et al., 2018). We choose this task because it also requires text comprehension;[1] the text sequences are shorter than in RC (a pair of single sentences); datasets are large; the complexity of the label space is small; and the label space furthermore does not vary between samples.

Second, rather than aiming to identify *natural language* inputs to probe a model's undersensitivity (as, e.g. achieved using Named Entity (NE) substitutions), we will here focus only on *deletions* of text, i.e. the removal of subsets of the given input tokens. This choice of perturbation space simplifies aspects of the formal verification procedure and follows prior work on undersensitivity (Feng et al., 2018), but generally does not produce natural language expressions as model inputs. An example of such an undersensitivity attack, concretely in the NLI task, can be found in Table 7.1.

Third, we only consider a single, non-state-of-the-art model: the decomposable attention model (DAM) architecture (Parikh et al., 2016). We focus on formally verifying a specification for this model in detail. The model comprises several types of the neural layers used in contemporary models, but uses non-contextualised pre-trained embeddings. While the formal verification of a larger NLP architecture such as BERT (Devlin et al., 2019) – which like DAM also utilises attention layers – is certainly worth pursuing, we choose the DAM architecture as a smaller initial step in the direction of formally verifying neural NLP models.

7.1.4 Chapter Overview

We will in this chapter describe a method to formally verify an undersensitivity specification on the DAM model. Concretely, we will verify that a model does not increase its prediction confidence under the removal of arbitrary subsets of input tokens. To this end we will leverage IBP, and we will then compare the verification

[1] In fact, NLI can be interpreted as a multiple-choice RC problem, where the question is whether the premise entails the hypothesis, and answer options correspond to the different NLI labels.

Premise (original)	*A little boy in a blue shirt holding a toy.*
Hypothesis	*A boy dressed in blue holds a toy.*
Prediction	Entailment (86%)
Premise (adversarially reduced)	*boy in a blue shirt*
Hypothesis	*A boy dressed in blue holds a toy.*
Prediction	Entailment (92%)

Table 7.1: Undersensitivity to input deletions in NLI: in this example the deletion of premise words increases model confidence, but the 'Entailment' label is not adequate any more. The issue was identified in prior work (Feng et al., 2018); here we intend to formally verify if any such adversarial reduction exists. Model: DAM (Parikh et al., 2016); dataset: SNLI (Bowman et al., 2015).

rates of several potential defence methods against undersensitivity. One particular defence we will investigate is the addition of an IBP-related auxiliary objective to the log-likelihood model training loss. We will see that this proves to be an overall moderately effective approach in directing models towards verifiably adhering to the given specification, yet largely more effective than other adversarial defence baselines. By furthermore measuring adversarial accuracy and the performance of an exhaustive verification oracle (which is possible for short sequences), we can derive additional insights about the verification process and possible bottlenecks.

List of Research Questions:

1. How can IBP be adapted to formally verify an undersensitivity specification on the DAM model?

2. How efficient, and how effective is IBP at verifying this specification?

3. How do different training methods aimed at defending against undersensitivity compare in terms of IBP-verification rates?

7.2 A Specification against Undersensitivity

We will next formulate a specification against undersensitivity with respect to input text deletions, and then introduce the methodology of specification verification, IBP in particular. Subsequently we will derive how IBP can be used to check for violations of the specification in the DAM architecture, and finally experiment with

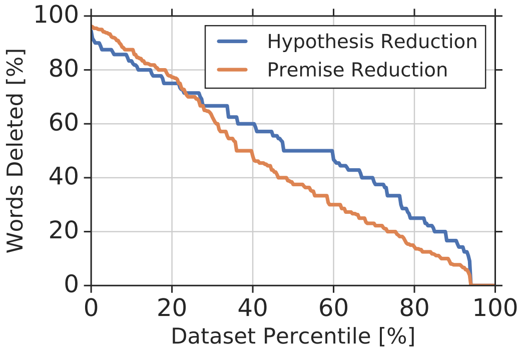


Figure 7.1: Fraction of words that can be deleted with an adversarial attack using beam search, for different percentiles of the dataset. Model: DAM; Dataset: SNLI (test).

this method to measure and compare the extent to which differently trained DAM models verifiably adhere to the undersensitivity specification.

In the previous chapter we observed that RC models do not reliably take into account entity information specified in the question. We will now consider adversarial *deletions* of input tokens. Prior work (Feng et al., 2018) has shown that adversarially deleting the majority fraction of input tokens can still leave the model prediction invariant, indicating that short lexical cues can be sufficient to trigger a model's decision on entailment. As neural models have the capacity to learn any association between input and output, the data distribution they are trained on determines how they achieve this. Models can learn to rely on shallow triggers – e.g. the presence of particular token combinations – to form their prediction. In the extreme case, even hypothesis-only baselines can succeed to a surprising degree in correctly predicting entailment (Poliak et al., 2018; Gururangan et al., 2018).

In Fig. 7.1 we plot the extent to which adversarial token deletions are possible in SNLI: we adversarially delete as many tokens as possible (following the beam search procedure of prior work (Feng et al., 2018)) from either the premise, or from the hypothesis, while the model increases its confidence for the same label prediction. This figure demonstrates that for many samples a substantial number of tokens can be removed. This holds true both for deletions from the premise, and to a very similar degree also for deletions of hypothesis tokens. For example, for 20% of the samples in the dataset, there exists a reduction of 78% (or more) of the premise words. That is, 22% or less of the premise tokens remain and still trigger the same prediction – with the same or higher model confidence than the original, unaltered input. For the scope of this entire chapter, we will limit our focus and methodological exploration to deletions of one of the two input sequences. We note that the DAM model structure is symmetric in its two input sequences, and that investigations into deletions of premise or hypothesis tokens can be pursued in a similar way.

In contrast to Chapter 6, where we substituted expressions, we now consider a perturbation space consisting of the inputs that have some of their input text removed. Concretely we consider arbitrary deletions of token combinations in one of the two sequences, and denote the resulting perturbation space derived from the original input x as $\mathcal{X}^{\text{in}}(x)$. Our specification then concerns the model probabilities for the label $\hat{y}(x)$, and the changes observed therein as fractions of input tokens are removed.

Concretely, our goal will be to check if there exists an input $z_0 \in \mathcal{X}^{\text{in}}(x)$ for which the model assigns a higher probability to $\hat{y}(x)$. We formalise this in the following specification:

$$\forall z_0 \in \mathcal{X}^{\text{in}}(x) : P(\hat{y}|z_0) \leq P(\hat{y}|x) \tag{7.1}$$

where $\hat{y} = \arg\max_y P(y|x)$ is the model's label prediction for the original input x.

7.2.1 Discussion

By enforcing the above Specification 7.1, we rule out undersensitivity attacks based on word deletions, i.e. $z_0 \in \mathcal{X}^{\text{in}}(x)$ for which $P(\hat{y}|z_0) > P(\hat{y}|x)$. This particular choice of specification is however not uncontroversial. Do we in fact *always* want to enforce that a model's probability for a label should remain the same or decrease, if some of its inputs are removed?

On the one hand, not all words carry the same relevance for determining entailment – some might not be relevant at all, such as stop words. On the other hand, some tokens alone are critical, and can single-handedly invert the correct entailment label of a given sentence pair, such as the token *"not"*. How exactly the output probabilities of a model should change as its inputs are gradually removed is hard to specify in detail, given the wide range of semantic variations in natural language expressions. Formulating a more detailed specification than the above is thus both cumbersome and prone to inconsistencies, though not an impossibility in principle.

Our aim here is to be careful about not constraining the model too much, while ruling out the possibility for the particular type of adversarial undersensitivity attack of increased model probabilities for altered inputs. The specification to *not increase* the output confidence, given arbitrary input subset removals, is a relatively conservative choice. We do, for example, not specify that probabilities should monotonically decrease as more and more words are removed. The specification is furthermore only about one of the labels: the one assigned to the original input.[2] Even with our fairly unrestricted choice of specification, we will see that Specification 7.1 can be positively verified only for the minority of samples in the DAM model.

A potential concern with the chosen specification is that it builds on the implicit assumption that the prediction of the original sample is correct. If however a model has low (nominal) accuracy, then the specification is about the model's confidence in mostly wrong predictions. This is not necessarily useful as a learning signal and may further deteriorate performance, as it can lower the relative priority of the main objective – fitting correct outputs to the given inputs. A second consideration is that enforcing the above specification may lead to models which generally favour less confident predictions for shorter inputs. This may bias a model towards fitting training samples with longer text sequences. Finally, a potential issue is that the specification is hard to justify for deletions of tokens capable of relativising the given statement (e.g. *"somewhat"*, *"maybe"*). Such tokens are however rare on the two datasets we will consider in our experiments, and erroneously enforcing the above specification for such cases is thus unlikely to be a major concern.

[2] Other, more task-specific alternatives are worth consideration – such as increasing the probability of the *neutral* label. We leave the exploration of such alternative specifications to future work, noting that much of the subsequent derivations in Section 7.5 can likely be recycled to this end.

7.3 Formal Verification of Neural Networks in NLP

7.3.1 Formal Model Verification

The central idea of formal model verification is to obtain a certifiable guarantee that a model adheres to a given specification. The specification is formulated mathematically, as a relationship between model inputs and outputs – our above stated Specification 7.1 is one such example. Note that rather than specifying desirable behaviour for concrete input-output pairs – such as annotated training examples, for which erroneous predictions are penalised and accuracy is measured – a specification in the broader sense aims to capture and relate specific properties in the input and output space.

A second important type of a model specification, which previous work has aimed to formally verify, is on model oversensitivity. These specifications address a model's robustness to adversarially chosen inputs from a given semantically invariant input perturbation space. For example they limit the amount of permissible output variation under l_∞-norm bounded perturbations of an input image (Gowal et al., 2018), or substitutions of synonyms in input text (Huang et al., 2019). The specification concept however is very broad and has, for example, been used also to guarantee adherence to the mechanical rules of physics for a model's outputs (Qin et al., 2019).

There are both *complete* and *incomplete* methods for formal model verification of model specifications; see Table 7.2 for a high-level summary of their verification properties. Both approaches can prove a model's full adherence to a given specification and rule out any violations, but they differ in whether there is a guarantee that they can always achieve this. Complete methods (Bunel et al., 2017; Cheng et al., 2017; Katz et al., 2017) are more computationally burdensome but can establish – with certainty – whether a specification is *not* satisfied. Incomplete methods on the other hand consider a relaxation of the verification problem (Weng et al., 2018; Wong and Kolter, 2018; Wang et al., 2018a; Raghunathan et al., 2018b). They are less expensive and more easily scalable, yet they cannot establish that where they fail to provide a certificate, there indeed exists a specification violation.

	establish presence	**verify non-existence**	***always* verify non-existence**
Adversarial Search	✓	×	×
Incomplete Verification	✓	✓	×
Complete Verification	✓	✓	✓

Table 7.2: Overview of capabilities with respect to adversarial attacks of complete and incomplete verification, as well as adversarial search. While adversarial search can identify adversarial samples / violations to a specification, it generally cannot verify their nonexistence. Incomplete verification methods can verify their nonexistence, but generally not for all samples. Complete verification methods can generally establish both the existence *and* nonexistence of violations.

To further clarify this: on samples where either incomplete or complete approaches do provide a certificate, the specification is verifiably satisfied. Where complete approaches provide no certificate, there definitely exists a specification violation. Where incomplete approaches do not provide a certificate, the model might still adhere to the specification, yet the verification method might just fail to provide such a certificate – hence their description as *incomplete*. Because of their superior efficiency, incomplete verification methods – such as IBP – can however be used during training, and direct models towards verifiably adhering to the given specification (Raghunathan et al., 2018a; Wong and Kolter, 2018; Dvijotham et al., 2018a; Gowal et al., 2018; Dvijotham et al., 2018b).

7.3.2 Verification of NLP Models

The verification of models in NLP has not received a lot of attention in the research community. While some work (Barr and Klavans, 2001) has considered the problem before the advent of deep learning in NLP, most recent work on neural model verification focuses on verification against adversarial attacks based on l_∞ norm-bounded adversarial image perturbations in computer vision. In contrast to the continuous image domain, inputs in NLP are discrete.

Neural network verification is a challenging problem, both in the computer vision and in the NLP setting, and both IBP and other methods have yet to be successfully scaled to deeper networks. Recent work in the NLP context (Huang et al., 2019; Jia et al., 2019) has used IBP to verify model behaviour against oversensitivity adversaries under synonym perturbations. Other work (Wang et al., 2019) has studied the verification of specifications regarding the length of a generative model's output sequence. In contrast, the type of specification we consider will be about model undersensitivity when removing combinations of input words, and we choose the attention-based DAM model as the object of our study.

7.3.3 Incomplete Verification using Interval Bound Propagation

Interval Bound Propagation (IBP) is an incomplete, yet very efficient verification method. From a technical viewpoint, it relies on a relaxation of the input perturbation space. Assuming vector inputs, IBP bounds the perturbation space via axis-aligned hyper-rectangles, tracks them throughout the network in the forward pass and bounds the corresponding activations in each layer, again using hyper-rectangles. IBP thus establishes bounds for all model activations, as well as the final model outputs associated with the input perturbation space. We will next describe this in more formal detail, partly adopting prior notational conventions (Gowal et al., 2018).

Concretely we assume that our model is a neural network that can be represented by a chain of functions h_k, indexed by $k \in \mathbb{N}$, which are either affine or elementwise nonlinear vector transformations. Let $\mathcal{X}^{\text{in}}(\mathbf{x})$ be a perturbation space derived from the original (vector) input $\mathbf{x}$.[3] Given an input $\mathbf{z}_0 \in \mathcal{X}^{\text{in}}(\mathbf{x})$, a network of K layers then computes its output by sequentially computing these individual transformations:

$$\mathbf{z}_k = h_k(\mathbf{z}_{k-1}) \quad k = 1, \dots, K \tag{7.2}$$

In a multi-class classification problem with C different classes – such as the one we will consider in the NLI task – the final model output $\mathbf{z}_K \in \mathbb{R}^C$ in the last layer comprises C different values, which correspond to the probabilities for each class.

For each layer k, IBP establishes an axis-aligned box (geometrically: a hyper-rectangle) that bounds the activation vector $\mathbf{z}_k$ from above and below. That is, the

[3] For notational convenience we identify the perturbation space $\mathcal{X}^{\text{in}}(\mathbf{x})$ (derived from a vector representation $\mathbf{x}$ of the input) with the previously used notion of $\mathcal{X}^{\text{in}}(x)$.

following relationship holds (elementwise):

$$\underline{\mathbf{z}}_k \leq \mathbf{z}_k \leq \overline{\mathbf{z}}_k \tag{7.3}$$

where $\underline{\mathbf{z}}_k$ defines a vector of elementwise lower, and $\overline{\mathbf{z}}_k$ a vector of elementwise upper bounds to $\mathbf{z}_k$. More concretely, for each individual neuron indexed by i (corresponding to the i'th dimension of the activation vector) the activation $z_{k,i}$ can be bounded from below and above as follows:

$$\begin{aligned}\underline{z}_{k,i} &= \min_{\underline{\mathbf{z}}_{k-1} \leq \mathbf{z}_{k-1} \leq \overline{\mathbf{z}}_{k-1}} \mathbf{e}_i{}^T h_k(\mathbf{z}_{k-1}) \\ \overline{z}_{k,i} &= \max_{\underline{\mathbf{z}}_{k-1} \leq \mathbf{z}_{k-1} \leq \overline{\mathbf{z}}_{k-1}} \mathbf{e}_i{}^T h_k(\mathbf{z}_{k-1})\end{aligned} \tag{7.4}$$

where $\mathbf{e}_i$ is the i^{th} standard basis vector of unit length. We will not derive in detail how these minima and maxima can be found for different types of transformation functions h_k, but refer the reader to the original work on IBP for this purpose. It is however worth pointing out that the above bounds can be computed efficiently for affine transformations and elementwise monotonic activation functions once bounds of the previous layer $\underline{\mathbf{z}}_{k-1}$ and $\overline{\mathbf{z}}_{k-1}$ are given. While this is trivial for monotonic elementwise activation functions, for affine transformations a key step is decomposing the weight matrix $\mathbf{W}^k = \mathbf{W}^k_+ + \mathbf{W}^k_-$ into one component $\mathbf{W}^k_+ = \max(\mathbf{W}^k, 0)$ of positive entries, and one component $\mathbf{W}^k_- = \min(\mathbf{W}^k, 0)$ containing negative entries. This effectively separates the weight matrix into two monotonic transformations, for which bounds can then more easily be computed.

Upper and lower bounds can thus recursively be computed for each layer based on the previous ones. In particular for the probability values in the last network layer, the resulting inequalities provide bounds for the model outputs of any $\mathbf{z}_0 \in \mathcal{X}^{\text{in}}(\mathbf{x})$. IBP hence establishes bounds on any, and notably also the worst-case violation of a given specification with input perturbation space $\mathcal{X}^{\text{in}}$. Concretely applied on the example of our Specification 7.1, if the upper bound $\overline{z}_{K,\hat{y}}$ of the last layer representations is less than the original output probability $P(\hat{y}|\mathbf{x})$, then Speci-

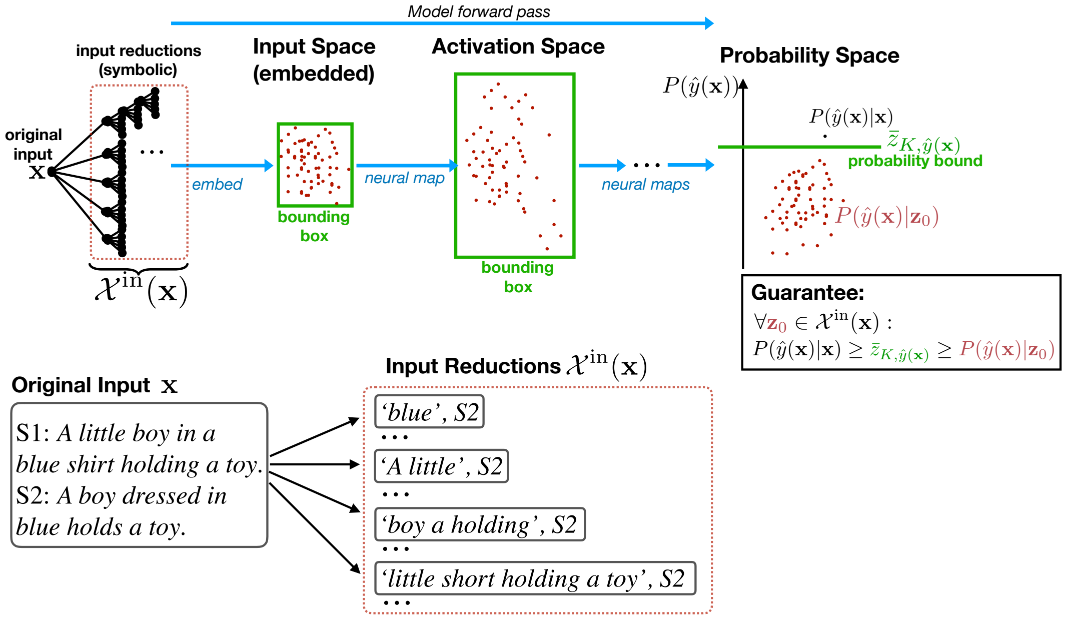

Model forward pass
input reductions (symbolic)
Input Space (embedded)
Activation Space
Probability Space
original input $\mathbf{x}$
embed
neural map
neural maps
bounding box
bounding box
$\mathcal{X}^{\text{in}}(\mathbf{x})$
$P(\hat{y}(\mathbf{x}))$
$P(\hat{y}(\mathbf{x})|\mathbf{x})$
$\bar{z}_{K,\hat{y}(\mathbf{x})}$
probability bound
$P(\hat{y}(\mathbf{x})|\mathbf{z}_0)$
Guarantee:
$\forall \mathbf{z}_0 \in \mathcal{X}^{\text{in}}(\mathbf{x})$:
$P(\hat{y}(\mathbf{x})|\mathbf{x}) \geq \bar{z}_{K,\hat{y}(\mathbf{x})} \geq P(\hat{y}(\mathbf{x})|\mathbf{z}_0)$
Original Input $\mathbf{x}$
S1: A little boy in a blue shirt holding a toy.
S2: A boy dressed in blue holds a toy.
Input Reductions $\mathcal{X}^{\text{in}}(\mathbf{x})$
'blue', S2
...
'A little', S2
...
'boy a holding', S2
...
'little short holding a toy', S2
...

Figure 7.2: Schematic overview: IBP for verifying an undersensitivity specification on a given sample $\mathbf{x}$, which consists of a pair of word sequences S_1 and S_2. Removing arbitrary token subsets of S_1 leads to an exponentially sized space of input reductions $\mathcal{X}^{\text{in}}(\mathbf{x})$. The representations of elements $\mathbf{z}_0 \in \mathcal{X}^{\text{in}}(\mathbf{x})$ (red) are bounded (green) in each network layer throughout the forward pass (blue). Bounds in probability space (on the right) can then be used to evaluate whether the undersensitivity specification is satisfied.

fication 7.1 is verifiably satisfied:

$$\forall \mathbf{z}_0 \in \mathcal{X}^{\text{in}}(\mathbf{x}) : P(\hat{y}|\mathbf{z}_0) \leq \bar{z}_{K,\hat{y}} \leq P(\hat{y}|\mathbf{x}) \tag{7.5}$$

By relying on Inequality 7.5, we can thus use the arising IBP bounds to check for any given $\mathbf{x}$, whether $\bar{z}_{K,\hat{y}} \leq P(\hat{y}|\mathbf{x})$. If this is satisfied, it allows us to infer that there exist no reduced input samples $\mathbf{z}_0 \in \mathcal{X}^{\text{in}}(\mathbf{x})$ with higher probability for $\hat{y}$ than $\mathbf{x}$, and we can thus certify that Specification 7.1 is satisfied. Fig. 7.2 gives a schematic overview of the overall approach.

In general, IBP bounds computed for the model outputs are loose, and even more so with more network layers. This is one of the reasons why we chose to verify the DAM model, which uses fewer neural layers than many of the more recent transformer-based NLP models. Because of the looseness of the bounds in the last layer, IBP tends to over-approximate the transformed image of $\mathcal{X}^{\text{in}}(\mathbf{x})$ in the output vector space. It uses these potentially very loose bounds, rather than the *actual* image of $\mathcal{X}^{\text{in}}(\mathbf{x})$ in the output space, and hence IBP is incomplete: it can fail to provide a verification certificate, even though the specification is satisfied.

IBP bounds can be efficiently computed in parallel, and with a similar time complexity as the standard forward pass.[4] IBP can be used during evaluation, to verify a previously trained model. It can however also be used during training by including a loss term that specifies how far the bounds in the last layer violate the specification (the extent to which $\bar{z}_{K,\hat{y}} > P(\hat{y}|\mathbf{x})$), and penalising the model correspondingly. While IBP has in the past been used for verifying oversensitivity specifications in models with feed-forward and convolutional layers (Gowal et al., 2018; Huang et al., 2019), we will next show how IBP can be applied on the DAM architecture, and use it to verify an undersensitivity specification.

7.4 The Decomposable Attention Model

We will next briefly recall the architecture of the DAM model (Parikh et al., 2016), largely adopting the notation of the original work. This serves both the purposes of refreshing our memory, and of introducing notation that we will subsequently rely on when describing the verification of Specification 7.1 for this model.

The DAM architecture was originally designed for the NLI task, and consists of several widely-used elements of neural NLP models, including word embeddings, attention, and feed-forward layers. It expects as input two token sequences, and its purpose is to predict one of three discrete classes, corresponding to different entailment labels. Rather than operating on sequences of token symbols, the DAM model takes as input sequences of input tokens that are already embedded as d-dimensional vectors, e.g. relying on pre-trained word embeddings such as *GloVe* (Pennington et al., 2014). Its input can thus be described as $\mathbf{A} = [\mathbf{a}_1; \ldots; \mathbf{a}_I] \in \mathbb{R}^{d \times I}$, and $\mathbf{B} = [\mathbf{b}_1; \ldots; \mathbf{b}_J] \in \mathbb{R}^{d \times J}$, where $[.;.]$ denotes concatenation, and I and J are the respective lengths of the two text sequences. The model then transforms individual word embedddings with a vector-valued function $F(.)$, and pairs of words (one in

[4] A ready IBP implementation can be found at `https://github.com/deepmind/interval-bound-propagation`, which we also use in our implementation.

each of the two sequences) are then compared and aggregated in a scalar score:

$$e_{ij} = F(\mathbf{a}_i)^T F(\mathbf{b}_j) \in \mathbb{R} \tag{7.6}$$

The vector function F can be any differentiable function to allow for gradient-based training, e.g. an MLP, or a linear transformation. We next adopt matrix notation across word position pairs (i, j) to summarise the above more concisely. Eq. (7.6) can then be reformulated as

$$\mathbf{E} = F(\mathbf{A})^T F(\mathbf{B}) \in \mathbb{R}^{I \times J} \tag{7.7}$$

The matrix $\mathbf{E}$ thus possesses one scalar term for each pair of words taken from the two input sequences. In normalising the cumulative exponentiated mass across either of its two axes (i, or j), the DAM model then computes two attention masks – one for each of the input sequences:

$$P_{ij}^{(\mathbf{A})} = \frac{\exp(e_{ij})}{\sum_k \exp(e_{kj})}; \;\; \mathbf{P}^{(\mathbf{A})} \in \mathbb{R}^{I \times J} \tag{7.8}$$

$$P_{ij}^{(\mathbf{B})} = \frac{\exp(e_{ij})}{\sum_k \exp(e_{ik})}; \;\; \mathbf{P}^{(\mathbf{B})} \in \mathbb{R}^{I \times J} \tag{7.9}$$

The coefficients of these two masks $\mathbf{P}^{(\mathbf{A})}$ and $\mathbf{P}^{(\mathbf{B})}$ now serve as contribution weights for a convex sum of the original input word vectors, and thus perform an attention-aggregation of the vectors in the two original sequences:

$$\begin{aligned} \mathcal{A} =& \mathbf{A} \cdot \mathbf{P}^{(\mathbf{A})} \in \mathbb{R}^{d \times J} \\ \mathcal{B} =& \mathbf{B} \cdot (\mathbf{P}^{(\mathbf{B})})^T \in \mathbb{R}^{d \times I} \end{aligned}$$

To summarise, the matrices $\mathcal{A}$ and $\mathcal{B}$ contain attention-weighted sums of word vectors – one for each position i and j of the two input sequences. These transformed input representations are then concatenated with the original input vectors from the same sequence, and transformed using a feed-forward network $G : \mathbb{R}^{2d} \to \mathbb{R}^{d'}$, applied on each sequence position (indexed by i, or j) individually. The resulting vectors are then aggregated with a sum over the resulting vector representations across all the positions of each of the two sequences. In short:

$$\mathbf{v}_1 = \sum_i G([\mathbf{a}_i; \mathcal{B}_i]) \in \mathbb{R}^{d'} \tag{7.10}$$

$$\mathbf{v}_2 = \sum_j G([\mathcal{A}_j; \mathbf{b}_j]) \in \mathbb{R}^{d'} \tag{7.11}$$

Finally, both $\mathbf{v}_1$ and $\mathbf{v}_2$ are concatenated and transformed using a vector-valued function (again, a feed-forward network) $H : \mathbb{R}^{2d'} \to \mathbb{R}^C$ that maps them onto logits for each of the C classes of the entailment task.

7.5 Verifying Undersensitivity Guarantees for the DAM Architecture

After having laid out the basic structure of the DAM model, our next goal is to formally verify Specification 7.1 for a given (vectorised) model input $\mathbf{x} = (\mathbf{A}, \mathbf{B})$. By establishing and checking bounds on even the most extreme output probabilities of any point in $\mathcal{X}^{\text{in}}(\mathbf{x})$, the model's behaviour on the entire perturbation space $\mathcal{X}^{\text{in}}(\mathbf{x})$ can be checked. Compared to prior work where IBP has been used to verify neural network models (Gowal et al., 2018; Huang et al., 2019), the DAM model verified here contains an attention component, which poses a challenge due to the difficulty of bounding altered attention normalisation mass. We describe how this challenge can be overcome by exploiting the fact that both the vectors in $\mathcal{A}$ and $\mathcal{B}$ are formed as convex sums and can hence be bounded by bounds on their unweighted summand constituents.

We will next describe in detail how Specification 7.1 can be verified for the DAM model using IBP. First, we will lay out how the model and its latent represenations change when deleting an individual word at a fixed and given position. Second, we will generalise this to the removal of individual words at an arbitrary position in the sequence. Third, we will relax this even further to the case of removing arbitrary combinations of tokens from the input.

7.5.1 Deleting Individual Words at a Particular Position

We begin by describing how the latent activations of the model change when a particular token at a given position r is deleted from one of the input sequences. We note that the DAM architecture is symmetric in its two sequences, and thus assume, without loss of generality, that we delete a word from the second sequence.

All vector and matrix representations resulting from altered inputs with partially deleted text will from here on be marked with a bar (as in $\bar{\mathbf{B}}$). For example, when deleting a given token with index r in the sequence:

$$\bar{\mathbf{B}} = [\mathbf{b}_1, \ldots, \mathbf{b}_{r-1}, \mathbf{b}_{r+1}, \ldots, \mathbf{b}_J] \in \mathbb{R}^{d\times(J-1)} \tag{7.12}$$

whereas on the other hand $\bar{\mathbf{A}} = \mathbf{A}$.

The transformation $F(.)$ is applied position-wise for each vector in the sequence. Consequently the deletion of a word remains isolated thus far, and

$$\bar{\mathbf{E}} = F(\bar{\mathbf{A}})^T F(\bar{\mathbf{B}}) \in \mathbb{R}^{I\times(J-1)} \tag{7.13}$$

still holds the same values as before, however the r-th column is removed. Similarly, the attention distribution $\bar{\mathbf{P}}^{(\mathbf{A})}$ contains the same values as $\mathbf{P}^{(\mathbf{A})}$ before, however again the r-th column is deleted. That is, for $i = 1,\ldots,I$ and $j = 1,\ldots,J$ such that $j \neq r$:

$$\bar{P}_{ij}^{(\mathbf{A})} = \frac{\exp(\bar{e}_{ij})}{\sum_k \exp(\bar{e}_{kj})};\ \ \bar{\mathbf{P}}^{(\mathbf{A})} \in \mathbb{R}^{I\times(J-1)} \tag{7.14}$$

The second attention distribution $\bar{\mathbf{P}}^{(B)}$, however, now holds renormalised values. These entries maintain their respective relative ordering, but their values increase due to the removal of the r-th summand in the normalisation denominator. Concretely, for $j \neq r$:

$$\bar{P}_{ij}^{(\mathbf{B})} = \frac{\exp(e_{ij})}{\sum_{k\neq r} \exp(e_{ik})};\ \ \bar{\mathbf{P}}^{(\mathbf{B})} \in \mathbb{R}^{I\times(J-1)} \tag{7.15}$$

That is, we can alternatively describe $\bar{P}_{ij}^{(\mathbf{B})}$ as its original value $P_{ij}^{(\mathbf{B})}$, rescaled using a renormalisation factor:

$$\bar{P}^{(\mathbf{B})}_{ij} = P^{(\mathbf{B})}_{ij} \cdot \frac{\sum_k \exp\left(e_{ik}\right)}{\sum_{k \neq r} \exp\left(e_{ik}\right)} \tag{7.16}$$

In summary, the attention distribution $P^{(\mathbf{B})}_{ij}$ mostly maintains its values when removing the word at position r, but they are renormalised to compensate for the contribution of the r-th word that is now missing. The DAM model next calculates $\bar{\mathcal{A}}$ and $\bar{\mathcal{B}}$. Again, the values of

$$\bar{\mathcal{A}} = \bar{\mathbf{A}} \cdot \bar{\mathbf{P}}^{(\mathbf{A})} \in \mathbb{R}^{d \times (J-1)} \tag{7.17}$$

are identical to before, and the r-th column is deleted. On the other hand, for

$$\bar{\mathcal{B}} = \bar{\mathbf{B}} \cdot (\bar{\mathbf{P}}^{(\mathbf{B})})^T \in \mathbb{R}^{d \times I} \tag{7.18}$$

the dimensions remain constant, but the values differ from the original values in $\mathcal{B}$ since $\bar{\mathbf{B}}$ contains fewer vector entries and $(\bar{\mathbf{P}}^{(\mathbf{B})})^T$ is re-scaled accordingly.

These modified quantities $\bar{\mathbf{A}}, \bar{\mathbf{B}}$ and $\bar{\mathcal{A}}, \bar{\mathcal{B}}$ are then further propagated through the remaining layers G and H to obtain a concrete model output. It is worth pointing out that all of the above expressions can be calculated in closed form, and there is no need to apply IBP yet.

7.5.2 Deleting Individual Words at Arbitrary Positions

So far we have described $\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathcal{A}}$ and $\bar{\mathcal{B}}$ in closed form, for a *specified* position r in the sequence. The resulting expressions can be calculated for the deletion of any word, i.e. for any sequence position r. When generalising the above to removals of individual tokens at *arbitrary* positions, we can calculate the min and max of these previously obtained expressions (elementwise) along the sequence, and thus obtain upper and lower bounds. For example

$$\bar{\mathcal{B}}_{\max} = \max_{r=1,\ldots,J} \bar{\mathcal{B}}(r) \tag{7.19}$$

$$\bar{\mathcal{B}}_{\min} = \min_{r=1,\ldots,J} \bar{\mathcal{B}}(r) \tag{7.20}$$

serve as upper and lower bounds for $\bar{\mathcal{B}}$. The original DAM model then proceeds in propagating these activations into the vector transformations G and H (cf. Eq. (7.10) and Eq. (7.11)), each a dense feed-forward model with two layers and a *ReLU* activation. We deviate from the original model in using a *softplus* activation, which serves as a smooth approximation with strictly positive values.[5]

Together with $\bar{\mathbf{A}} = \mathbf{A}$, the above bounds on $\bar{\mathcal{B}}$ are propagated through G using IBP, hence establishing bounds on $\bar{\mathbf{v}}_1$. On the other hand, $\bar{\mathbf{v}}_2$ can be calculated from $\mathbf{v}_2$, by subtracting the contribution of its r-th summand (for fixed deletions at position r, cf. Eq. (7.11)). Extending this to arbitrary r, the subtracted vector can be bounded from above and below using $\max_{r=1,\ldots,J}\{G([\bar{\mathcal{A}}_j;\bar{\mathbf{b}}_j])\}$ and $\min_{r=1,\ldots,J}\{G([\bar{\mathcal{A}}_j;\bar{\mathbf{b}}_j])\}$.

Given these bounds on $\bar{\mathbf{v}}_1$ and $\bar{\mathbf{v}}_2$, IBP is then used to propagate them through the feed-forward layers in H. Consequently, we obtain bounds on the logits for each class – and thus output probabilities – which bound the model outputs for any arbitrarily chosen deletion of a single input word.

7.5.3 Deleting Multiple Tokens at Arbitrary Position

Thus far we have derived how the model behaviour changes – and can be bounded – when removing single tokens at an arbitrary position. We next generalise this to the deletion of an arbitrary subset of input tokens (in one of the two sequences). We will denote the indices of the remaining tokens as $D \subsetneq \{1,\ldots,J\}$ with $D \neq \emptyset$. In this case again, the entries of any remaining word vectors $\mathbf{a}_i$ and $\mathbf{b}_j$ ($j \in D$) are unaltered.

[5] This ensures strict monotonicity of $\mathbf{v}_1$ (and $\mathbf{v}_2$) in the number of summands, and thus a strictly positive difference between $\mathbf{v}_1$ and $\bar{\mathbf{v}}_1$. Without this guarantee, there could be j such that $G([\bar{\mathcal{A}}_j;\bar{\mathbf{b}}_j]) = 0$, thus potentially resulting in $\bar{\mathbf{v}}_1 = \mathbf{v}_1$. This would map the perturbed input onto the same output as the original, thus leading to equality in Specification 7.1.

Bounding $\bar{\mathbf{v}}_1$: Recall that for the original, unaltered input (cf. Eq. (7.10)):

$$\mathbf{v}_1 = \sum_i G([\mathbf{a}_i;\mathcal{B}_i]) \in \mathbb{R}^{d'}$$

If we were in possession of bounds on $\bar{\mathbf{a}}_i$ and $\bar{\mathcal{B}}_i$, then upper and lower bounds for $\bar{\mathbf{v}}_1$ could be computed directly using IBP for G and the sum. The entries of $\mathbf{a}_i$ do not change when deleting tokens from the second sequence ($\bar{\mathbf{a}}_i = \mathbf{a}_i$). Thus we will focus on deriving bounds for $\bar{\mathcal{B}}_i$.

Recall that during the attention aggregation, the columns in $\mathcal{B}$ are calculated as a convex sum of the column vectors in $\mathbf{B}$. That is, the i^{th} column of $\mathcal{B}$ is calculated as $\mathcal{B}_i = \sum_j \mathbf{b}_j P^{(B)}_{i,j}$, where $P^{(B)}_{i,j}$ corresponds to the entry of $\mathbf{P}^{(\mathbf{B})}$ at position (i,j). Consequently, the largest and smallest values that the entries of $\bar{\mathcal{B}}_i$ can possibly take can be bounded by the elementwise minima and maxima of individual column vectors in $\mathbf{B}$:

$$\mathbf{b}_{min} = \min_{j=1,\dots,J} \{\mathbf{b}_j\} \in \mathbb{R}^d \tag{7.21}$$

$$\mathbf{b}_{max} = \max_{j=1,\dots,J} \{\mathbf{b}_j\} \in \mathbb{R}^d \tag{7.22}$$

The values of these vectors form elementwise bounds on $\bar{\mathcal{B}}_i$, e.g. $\mathbf{b}_{max}$ from above:

$$\bar{\mathcal{B}}_i = \sum_{j\in D} \mathbf{b}_j \cdot \bar{P}^{(\bar{B})}_{ij} \leq \sum_{j\in D} \mathbf{b}_{max} \cdot \bar{P}^{(\bar{B})}_{ij} = \mathbf{b}_{max} \cdot \sum_{j\in D} \bar{P}^{(\bar{B})}_{ij} = \mathbf{b}_{max} \cdot 1 = \mathbf{b}_{max} \tag{7.23}$$

It is worth pointing out that no matter which tokens are deleted, the adjusted attention distribution $\bar{P}^{(\bar{B})}_{i,j}$ always sums to 1. A similar relationship follows for $\mathbf{b}_{min}$ when bounding $\bar{\mathcal{B}}_i$ from below.

Bounding $\bar{\mathbf{v}}_2$: Next, we recall that for the original, unaltered input, (cf. Eq. (7.11)):

$$\mathbf{v}_2 = \sum_j G([\mathcal{A}_j;\mathbf{b}_j]) = \sum_j \mathbf{g}_j \in \mathbb{R}^{d'} \tag{7.24}$$

where we introduce the expression $\mathbf{g}_j = G([\mathcal{A}_j;\mathbf{b}_j])$ for subsequent notational convenience. The vector-valued function G of the DAM model is a two-layer feed-forward network using a ReLU activation function. Thus all entries of $\mathbf{g}_j$ are strictly non-negative ($\mathbf{g}_j \geq 0, \forall j$). Since we slightly deviate from the original model in using a *softplus* activation, the entries of $\mathbf{g}_j$ are strictly positive ($\mathbf{g}_j > 0, \forall j$). Consequently the sum $\sum_j \mathbf{g}_j$ decreases monotonically in each vector entry when some of the summands $\mathbf{g}_j$ are removed – or, conversely, increases monotonically when new summands are included. We will consider the two most extreme cases of removing arbitrary combinations of tokens: i) deleting all tokens but one ii) deleting exactly (and only) one token. The values of $\bar{\mathbf{v}}_2$ will then be bounded for any combination of removed words that can be situated between these two extremes.

In the first of the two cases, where all tokens but one, at position r, are deleted, the expression remaining is $\bar{\mathbf{v}}_2 = \mathbf{g}_r$. Computing the elementwise minimum across all possibilities for r can thus provide us with a lower bound on *any* $\bar{\mathbf{v}}_2$; not only for deletions of all but one word, but for any other deletion of fewer words as well, due to the strictly positive entries of each g_j:

$$\bar{\mathbf{v}}_2 \geq \min_{r=1,\dots,J} \{\mathbf{g}_r\} \tag{7.25}$$

In the second of the two cases, where merely one token is deleted from the input (at position r), only one summand is subtracted from the original $\mathbf{v}_2$: $\bar{\mathbf{v}}_2 = \mathbf{v}_2 - \mathbf{g}_r$. We can thus bound $\bar{\mathbf{v}}_2$ using the (elementwise) smallest value that $\mathbf{g}_r$ can assume, for any r.

$$\bar{\mathbf{v}}_2 = \mathbf{v}_2 - \mathbf{g}_r \leq \mathbf{v}_2 - \min_{r=1,\dots,J} \{\mathbf{g}_r\} \tag{7.26}$$

Since $\bar{\mathbf{v}}_2$ increases monotonically with its number of summands (i.e. the number of tokens in the input), any further deletion of tokens will only further decrease the value of the entries of $\bar{\mathbf{v}}_2$. Hence the above expression provides us with an upper bound for arbitrary combinations of removed tokens. In summary, we obtain the following upper and lower bounds for $\bar{\mathbf{v}}_2$:

$$\min_{r=1,\dots,J} \{\mathbf{g}_r\} \leq \bar{\mathbf{v}}_2 \leq \mathbf{v}_2 - \min_{r=1,\dots,J} \{\mathbf{g}_r\} \tag{7.27}$$

Having established these bounds for both $\bar{\mathbf{v}}_1$ and $\bar{\mathbf{v}}_2$, we can then proceed and propagate them using IBP through the two-layer MLP H that follows $\mathbf{v}_1$ and $\mathbf{v}_2$ in the forward pass of the DAM architecture.

This concludes our derivations on bound propagation for the deletion of arbitrary combinations of input tokens. Given arbitrary inputs x (or, in vectorised form: $\mathbf{x}$), we can compute bounds on the the output probabilities the DAM model assigns to any reduced input $\mathbf{z}_0 \in \mathcal{X}^{\text{in}}(\mathbf{x})$ and establish whether Specification 7.1 is satisfied. We will next proceed to investigating this experimentally.

7.6 Experiments

7.6.1 Datasets

To what extent can the above described IBP-based method verify that a trained DAM model adheres to the undersensitivity Specification 7.1? We will next experimentally evaluate the method, and compare different training approaches on two NLI datasets: SNLI (Bowman et al., 2015) and MNLI (Williams et al., 2018). Both follow the same input-output schema: an input of two symbol sequences has to be classified according to a catalogue of three possible labels: *Entailment*, *Contradiction*, and *Neutral*.

Prior work (Feng et al., 2018) has established undersensitivity to the removal of hypothesis tokens for SNLI. We here investigate the phenomenon also for MNLI, and show that it holds for deletions of premise words. In all the subsequent experiments we delete tokens from the premise, but note that the undersensitivity phenomenon is present to a similar extent in both deletions of premise and hypothesis tokens (cf. Fig. 7.1), and that similar investigations could be undertaken for hypothesis reductions, given the symmetric structure of the DAM architecture.

For experiments on the SNLI dataset we use the given standard splits and tune hyperparameters on the validation set while reporting experimental outcomes on the test set. For experiments on the MNLI dataset, we split off 2000 samples from the validation dataset for development purposes, using the remaining examples to compute the test metrics. Overall we closely follow the experimental setup of the original work introducing the DAM architecture (Parikh et al., 2016), with the same types of feed-forward components, layer size, dropout, and word embedding hyperparameters.

7.6.2 Evaluation Metrics

In our experiments we will measure model behaviour according to the subsequent list of metrics, all evaluated on the respective test set:

1. **Accuracy:** The standard classification accuracy used in prior work on this task, i.e. the fraction of test samples for which the prediction is correct.

2. **Verified Accuracy:** The fraction of test samples for which the prediction is correct *and* we can verify, using the above described verification procedure, that Specification 7.1 is satisfied. Note that standard accuracy is an upper bound for this metric.

3. **Robustness to Undersensitivity Attacks *("Beam Search Heuristic")*:** Here we utilise beam search in the same way as prior work (Feng et al., 2018) in an attempt to identify violations to Specification 7.1 in the perturbation space. We begin the search with the full input sequence and delete tokens step by step while maintaining a beam of width 10. We then measure, concretely, whether our search has *not* yielded a counterexample *and* whether the model prediction was accurate. Note that since this search heuristic does not exhaustively cover the input perturbation space, we can miss violations to Specification 7.1. The resulting metric establishes an upper bound for the above described *verified accuracy* metric; conversely *verified accuracy* is bounded from below by the *beam search heuristic* metric.

7.6.3 Training Methods

We will first evaluate the standard log-likelihood model training, and then proceed in investigating to what extent established defence methods against adversarial attacks can increase the rate of positive verifications of the undersensitivity specification. Concretely, we will compare the subsequent list of training approaches:

1. **Standard Training:** Here we follow the standard log-likelihood training commonly used in discriminative classification tasks.

2. **Data Augmentation:** We have previously in Chapter 6 found that the augmentation of the training data with samples from the perturbation space can improve the model's robustness to adversarial attacks. Here, we add such training samples with randomly deleted subsets of words (using a Bernoulli probability of 50% for retaining any of the given tokens, and sampling them independently of one another). If such a randomly drawn reduced input example $\mathbf{z}_0 \in \mathcal{X}^{\text{in}}(\mathbf{x})$ results in a model probability $P(\hat{y}(x)|\mathbf{z}_0) > P(\hat{y}(x)|\mathbf{x})$, then we calculate the (positive) difference between them, multiply it with a scalar $\lambda > 0$ and add it to the loss. Thus, during training we penalise the DAM model whenever it violates the undersensitivity specification as determined from randomly drawn samples in $\mathcal{X}^{\text{in}}(\mathbf{x})$. In this procedure, randomly reduced samples are drawn in the same proportion as original samples.

3. **Adversarial Training:** While in the previously described defence method we relied on randomly subsampled sequences of input words, we now systematically search for deletions of word combinations that most strongly violate the undersensitivity specification. We consider two types of adversarial search: i) sampling 512 elements from $\mathcal{X}^{\text{in}}(\mathbf{x})$ at random, and picking the strongest violation ii) beam search of width 10, again using the procedure laid out in prior work (Feng et al., 2018). In our search we identify those inputs with the largest gap between the output probability $P(\hat{y}(x)|\mathbf{x})$ of the original input, and the probability for the same output using the reduced input $P(\hat{y}(x)|\mathbf{z}_0)$, i.e. the largest violation to the specification 7.1. The degree of violation (where it is larger than 0) is multiplied by $\lambda > 0$ and added as a contribution to the training loss, in the same way as for data augmentation. The perturbed inputs are continuously re-sampled or searched for throughout the whole training procedure, ensuring that adversarial samples are always up-to-date with the most recent iteration of the model parameters.

4. **Entropy Regularisation:** Prior work (Feng et al., 2018) has identified entropy regularisation (computed on the distribution of output probabilities) as a modification to standard training that can help alleviate model undersensitivity. We hence include it in our comparison and investigate to what extent it can also improve the verification rates compared to standard training.

5. **IBP-Training:** Finally, we experiment with the addition of an auxiliary training objective that is directly derived from the IBP verification procedure. IBP establishes upper bounds on output probabilities for the entire reduction space $\mathcal{X}^{\text{in}}(\mathbf{x})$ (see Eq. (7.5)). If the bound $\bar{z}_{K,\hat{y}}$ for the probability of $\hat{y}$, however, exceeds the original probability $P(\hat{y}|\mathbf{x})$ for the predicted class $\hat{y} = \arg\max_y P(y|\mathbf{x})$, a positive difference between them emerges: $\Delta(\mathbf{x},\hat{y}) = \bar{z}_{K,\hat{y}} - P(\hat{y}|\mathbf{x})$. We use this expression $\Delta(\mathbf{x},\hat{y})$ to define a hinge loss (ensuring its contribution to the loss only where it is indeed positive) that we multiply with $\lambda > 0$ and add to the original training objective. Note that this bound $\bar{z}_{K,\hat{y}}$ bounds the output probability of any adversarially chosen (or randomly sampled) input $\mathbf{z}_0 \in \mathcal{X}^{\text{in}}(\mathbf{x})$. The value of $\bar{z}_{K,\hat{y}}$ over-approximates the model output probability for any sample arising from arbitrary removals of tokens in the input, and thus covers the entire perturbation space.

7.6.4 Training Details

Most of the aforementioned training approaches utilise an auxiliary contribution to the log-likelihood training loss. This contribution is additive and computed for the same batch of SGD samples as the log-likelihood part of the objective. To identify a viable degree of contribution to the loss we tune the scalar hyperparameter $\lambda \in \{10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}\}$ that scales the contribution of the auxiliary objective to the loss.[6] Our experiments all use an initial learning rate of 10^{-3}, optimisation using Adam, and batch size 128. Early stopping is conducted based on *verified accuracy*, halting at a maximum of 3M training steps.

When performing *IBP-Training*, it proves useful to phase in the perturbation

[6] Naturally, this is done separately for each of the different training approaches.

space volume continuously, i.e. not begin with its full size but gradually increase the volume to its full extent (Gowal et al., 2018). This allows for the gradual adaptation of an otherwise potentially very strong loss signal that can over-rule any signal stemming from fitting the original training data. We achieve this by calculating – for each dimension of the input – the arithmetic mean of the upper and lower bound. We then slowly phase in the contribution of the IBP-derived loss contribution by linearly interpolating from this dimension-wise arithmetic mean to the actual upper and lower bounds: beginning with a volume of 0, we linearly increase the volume until it reaches its full extent and covers the entire volume of the perturbation space. We tune the interval allowed for this gradual phasing in of the volumne in $\{10^0, 10^3, 10^4, 10^5, 10^6\}$ training steps. This approach also allows for linearly inflating the perturbation volume to an extent larger than the full size, which could potentially result in larger margins $\Delta(\mathbf{x}, \hat{y}) = \bar{z}_{K,\hat{y}} - P(\hat{y}|\mathbf{x})$ and improved verification levels. We briefly experimented with this, but did not observe improved verifiability outcomes as a result.

7.6.5 Results: General Observations

Results of our experiments are given in Tables 7.3 and 7.4 for the two datasets, respectively. Our first observation is that indeed, for a small, yet non-negligible fraction of samples the undersensitivity specification can be positively verified using IBP-based verification – across all training methods. The difference between standard accuracy, and verified accuracy is however striking: compared to all the samples that have been predicted correctly, only a small fraction IBP-verifiably adheres to the undersensitivity specification.

Next, we observe that beam search attacks succeed in almost all cases, leaving for the standard training baseline only 3.36% and 8.77% of samples correctly answered without successful attack, on the two datasets respectively. It is at this point worth pointing out that these adversarially reduced samples pose real challenges to humans attempting to predict the correct entailment label. Prior work (Feng et al., 2018) has conducted an analysis of adversarial undersensitivity attacks concretely on SNLI, which were identified using the beam search procedure also used here.

Training Method	**Accuracy**	**Verified Accuracy**	**Beam Search Heuristic**
Standard Training	77.22	2.83	3.36
Data Augmentation	76.37	5.09	6.27
Adversarial Training: random	76.89	1.79	4.16
Adversarial Training: beam search	76.09	5.48	**23.76**
Entropy Regularisation	**77.32**	5.82	6.28
IBP-Training	75.51	**18.36**	19.26

Table 7.3: Experimental outcome for different training methods on the SNLI dataset, as measured in standard accuracy, verified accuracy (using IBP verification), and the beam search heuristic, which uses adversarial search to find violations to the undersensitivity specification. Models are selected based on verified accuracy; all numbers are in [%].

This study found that humans asked to predict the label of adversarially reduced samples have considerable drops in their predictive accuracy compared to the original samples: -48.7%, -2.7% and -15.6% (absolute differences) for *'entailment'*, *'neutral'*, *'contradiction'*, respectively (cf. Table 1 in (Feng et al., 2018)).

In our experiments, we next observe that there are substantial differences in verified accuracy *between* training methods, and notably also improvements over the verification rates achieved using standard training. In particular the addition of a dedicated IBP-related objective in IBP-Training leads to a sharp increase in verified accuracy to 18.36% and 17.44% for the two datasets, respectively.

The improved IBP-verifiability however comes at a price: the standard accuracy on the test set is diminished on both datasets; not by much for SNLI, but drastically on the MNLI dataset. The standard accuracy of the different training approaches is also diminished in comparison to the originally reported test accuracy (Parikh et al., 2016) as we tune for high IBP-verification rates on the development set.

These observations suggest that IBP-verifiability and standard test accuracy may be – at least partially – at odds, in a similar way as previously noted for standard accuracy and adversarial accuracy (Tsipras et al., 2019). The inclusion of a new component to the objective function, which is simultaneously fitted with the log-likelihood objective – as well as model selection based on the corresponding *verified accuracy* metric – ultimately lead to lower standard accuracy values.

Training Method	Accuracy	Verified Accuracy	Beam Search Heuristic
Standard Training	60.00	7.77	8.77
Data Augmentation	**62.02**	1.93	4.26
Adversarial Training: random	61.89	2.60	5.04
Adversarial Training: beam search	58.74	0.45	7.44
Entropy Regularisation	60.74	8.83	9.47
IBP-Training	44.95	**17.44**	**19.07**

Table 7.4: Experimental outcome for different training methods on the MNLI dataset, as measured in standard accuracy, verified accuracy (using IBP verification), and the beam search heuristic, which uses adversarial search to find violations to the undersensitivity specification. Models are selected based on verified accuracy; all numbers are in [%].

This is however not necessarily a surprising result: the undersensitivity specification enforces a much stronger modelling requirement for individual samples than merely predicting the correct one of three labels: it constrains the latent activations of arbitrary input reductions. The predictive success of standard NLI models may to a large extent be based on spurious cues – as witnessed in the strong performance of hypothesis-only baselines in NLI (Poliak et al., 2018; Gururangan et al., 2018), which corresponds to an extreme case of deleting all premise tokens. By enforcing the undersensitivity specification, a model is hindered from forming high output probability values based on partial inputs, thus lowering data fit as well as train and test accuracy. We note that nominal test accuracy on an in-distribution test set does not directly reflect a model's susceptibility to undersensitivity attacks; it leaves the modeller blind to shortcomings such as relying on the hypothesis alone, and can thus only be seen as one (imperfect) indicator for a model's NLI capabilities.

7.6.6 Comparison of Training Methods

When comparing the different training methods, we first observe that standard training does only for a small fraction of samples lead to successful verification with IBP, both on SNLI and MNLI. Entropy regularisation has in prior work been shown to modestly improve adversarial vulnerability (Feng et al., 2018). Our experiments confirm this prior result with modest improvements in the *beam search heuristic* metric, and also in terms of verified accuracy.

The approaches that add reduced samples during training (data augmentation, as well as the two adversarial training approaches) lead to diverging results on the two datasets, yet where improvements in terms of verified accuracy are observed, they lack far behind those observed with IBP-based training. A possible explanation for the divergent results is the length of the sequences: in MNLI the premise symbol sequences are 6.2 tokens longer, on average. Consequently the reduction space is substantially larger (note the exponential growth of the perturbation space depending on the sequence length). Covering this space by selecting individual reduced samples (either at random, or using adversarial search) is thus less likely to find worst-case violations. Adversarial training with the beam search attack then leads to increased verification rates on SNLI but not MNLI, where the sequences are longer, and the worst case violation is less likely to be found.

It is worth pointing out that adversarial training on MNLI is overall not very effective, and leads to no improvements in either *verified accuracy* or *beam search heuristic*. On the other hand for SNLI, where sequences are shorter, both data augmentation and beam search adversarial training substantially improve the *verified accuracy* and *beam search heuristic* metrics. Finally, as mentioned before, IBP-training increases verified accuracy considerably and more effectively than any of the other approaches.

Recall that *verified accuracy* is bounded from above by the *beam search heuristic*. A gap between the two can arise for one of the following reasons: i) IBP is incomplete, i.e. the bounds it uses are too loose to effectively verify the specification, even though it actually holds true ii) the adversarial attack in the *beam search heuristic* misses violations to the specification due to it only partially covering the full search space. To investigate this further, we consider the model's performance on inputs with short sequences (at most 12 tokens) of the SNLI test set. For these sequences, exhaustively covering the full space arising from deleting arbitrary sequences of input tokens is computationally feasible, with at most 2^{12} forward passes per sample. In Table 7.5 we compare the different training approaches on these samples in terms of verified accuracy (computed with IBP, which is incomplete), verified accuracy (computed with exhaustive verification, which is complete), and,

	Verified Accuracy				
Training	**BSH**	**Exhaustive**	**IBP**	**Exh./BSH**	**IBP/Exh.**
Standard Training	5.37	5.13	4.34	95.5%	84.6%
Data Augmentation	8.78	8.59	6.48	97.8%	75.4%
Adversarial:random	7.03	6.88	1.87	97.9%	27.2%
Adversarial:beam	**32.14**	**31.90**	5.13	99.3%	16.1%
Entropy Regul.	9.28	8.90	8.35	95.9%	93.8%
IBP-Training	20.94	20.68	**19.29**	98.8%	93.2%

Table 7.5: Comparison of different verification-related metrics for SNLI inputs with up to 12 tokens. BSH: Beam Search Heuristic; Exh.: Exhaustive. The ratios in the two latter columns indicate how many cases BSH and IBP miss: the difference to 100% in each column shows how many adversarial samples BSH misses, and how many actually verifiable cases IBP fails to verify, respectively.

again, the beam search heuristic.

The verified accuracy values computed using exhaustive search (which is complete), is then bounded twofold by the other two metrics. On the one hand, it is bounded from below by the verified accuracy computed using IBP; the gap indicates the degree to which IBP verification is incomplete, i.e. fails to provide a verification certificate even though the specification is satisfied. On the other hand, the beam search heuristic metric provides an upper bound to verified accuracy computed using exhaustive search; the gap here indicates to what extent the beam search heuristic misses violations to the given specification due to its incomplete coverage of the search space – which exhaustive search can catch.

A first observation is that the verified accuracy rates on this subset of the SNLI test set are slightly higher than for the full test set. This indicates that verifying the undersensitivity specification is to some degree less challenging for shorter sequences.

Second, we observe that for all training approaches apart from adversarial training (rows 3 and 4), IBP-based verification indeed approaches the verification levels of the exhaustive search oracle: in each of these training approaches more than 75% of actually verifiable cases are positively verified. This means that the – in absolute terms – low verification rates we observe with IBP are not primarily due to excessively loose bounds that prevent IBP from verifying samples that in fact adhere to the specification, but indeed due specification violations. To clarify this further: for the large majority of samples it is not the incompleteness of IBP verification that leads to low IBP verification rates, but the lack of adherence of the model to the undersensitivity specification, as indicated by the comparative ratio with the exhaustive verification measure.

Third, adversarial training (of both types tested) leads to a very different outcome: here (and only here) there is a wide gap between the verification rates observed with IBP verification, and those theoretically possible using exhaustive search (27.2% and 16.1% IBP/Exh. ratio).

Fourth, the *beam search heuristic* (BSH) metric is generally closer to exhaustive verification (approximating it from above), than is IBP verification (approximating it from below). That is, IBP fails to verify more verifiable cases than BSH misses adversarial attacks, and the difference is most striking in the adversarial training baselines. That is, while effective at improving undersensitivity, adversarial training leads to wider IBP bounds – as seen in low IBP-based verification rates even where samples do adhere to the specification (i.e. where exhaustive verification succeeds).

These observations give a good indication for the situation on short sequences, yet due to the computational infeasibility of computing exhaustive attacks for larger input sequences it remains unclear how observations on the gaps between metrics translate to larger sequences.

7.6.7 Computational Efficiency of IBP Verification

As previously indicated, the computational burden that comes with exhaustively covering an exponentially large search space is substantial, rendering full verification impractical for larger input sequences. In Table 7.6 we compare the number of forward passed necessary to perform verification, both for the theoretical worst case, as well as an empirically computed bound. The incomplete and bound-based verification using IBP incurs only little additional cost of $O(1)$. This cost stems from computing upper and lower bounds for the activations, which is similar in complexity to a standard forward pass, and can be achieved in parallel (Gowal et al., 2018). Since a full exhaustive search is not possible, we instead consider exhaustive search

Metric	**Time[s]**	**# Eval's /sample**
Accuracy	2	1
IBP Verification	3	≈ 2
Exhaustive Search	–	2^L
Exh. Search up to 200K	45,674	200,000
Beam Search	505	$\approx b \cdot L$

Table 7.6: The computational cost of verifying the given specification. Left: time required (in [s]) to evaluate 300 randomly chosen SNLI samples, without cross-sample batching. Right: the theoretical worst-case for the number of forward passes necessary. L: sequence length; b: beam width.

up to a maximum of 200,000 forward passes, per example (which under-estimates the true computational cost).

Although only considering a bounded search with a potentially very long tail, and although the search halts once any violation to the specification has been found, we observe that both the theoretical worst case number of forward passes required – as well as the actual computation time in exhaustive evaluation – are orders of magnitude beyond IBP-based verification. This underscores the computational efficiency of IBP for verification.

7.7 Discussion

Our experiments demonstrate that it is possible to positively verify a specification addressing a model's undersensitivity using IBP. This specification is, however, only for a minority of samples verifiably satisfied. Disregarding the outcome of adversarial training, our results from Table 7.5 furthermore indicate that excessively loose IBP bounds are not the underlying cause of low verification rates, but rather the actual presence of specification violations.

Relying on IBP-based training substantially improves the verification rates, albeit at the cost of deteriorated nominal test accuracy. One possible interpretation of this result is that when enforcing the specification during training, models cannot rely as much on shallow surface cues any more to form their prediction. It might however also indicate that the bounds propagated are unnecessarily loose and impede the process of data fitting. Other methods for neural network verification – which bound the propagated perturbation spaces in other ways than with axis-parallel hyperrectangles (as does IBP) – may improve the "false negative" ratio associated with loose bounds, and impose less of an impediment to fitting the data.

Formal verification can offer a stronger guarantee than adversarial robustness to an adversarial attack: it can guarantee that no attack can succeed in breaking the specification, even with the strongest of possible adversaries. Evaluating adversarial accuracy might be conceptually easier than IBP-based verification, but it comes with increased computational overhead compared to evaluating IBP-based verified accuracy, and lacks a guarantee that a stronger adversary or bigger search budget might not uncover a valid or stronger attack. Prior work (Uesato et al., 2018) has discussed the problem of measuring adversarial robustness against weak adversaries, which under-estimates the true extent of a vulnerability, and may hence lead to flawed conclusions about a lack thereof.

We see IBP-based verification as a compromise between evaluating a model's robustness using adversarial search – which cannot uncover all violations – and using exhaustive search – which is computationally infeasible due to its exponential worst-case cost. In particular for larger reduction spaces, where statistical coverage is harder to achieve and exhaustive verification impossible, IBP-based verification can be a useful tool to efficiently evaluate a model's vulnerability, although at the cost of potential false negatives.

The fact of overall low absolute verification rates, as well as the large number of specification violations are a reason for concern about the model's robustness but not new; such observations are common for adversarial attacks on other tasks and have been related to datasets with high sample complexity (Schmidt et al., 2018).

7.8 Conclusion

Undersensitivity is a prevalent issue in both Reading Comprehension and Natural Language Inference. In this chapter we have explored a new method to both evaluate and potentially mitigate a model's undersensitivity to deletions in its input text.

We summarise answers to our initially posed list of research questions as follows:

1. **How can IBP be adapted to formally verify an undersensitivity specification on the DAM model?** We have described how bounds on perturbed inputs – resulting from partial input deletions – can be computed for every layer of the DAM architecture. The resulting bounds in probability space overestimate the probabilities of any perturbed input from an exponentially sized reduction space, and can thus be used as a proxy to certify that the probability of the original input is not exceeded by any perturbed input.

2. **How efficient, and how effective is IBP at verifying this specification?** IBP requires the propagation of bounds through the network, which comes at a constant cost in terms of number of forward passes; this stands in contrast to exhaustive verification, which requires an exponential number of forward passes. In a computationally feasible setup, which allows for exhaustive verification, we observe that IBP is an effective verification tool compared to the exhaustive verification oracle: IBP can certify 84.6% / 93.2% of the cases of the oracle, for a standard model and IBP-trained model, respectively.

3. **How do different training methods aimed at defending against undersensitivity compare in terms of IBP-verification rates?** All training methods tested have low IBP-verification rates, in absolute terms. Entropy regularisation can modestly improve IBP-verification rates over standard training (+2.99% and +1.06% on SNLI and MNLI), whereas data augmentation and adversarial training show diverging results for the SNLI and MNLI datasets, albeit at similarly low overall rates. Training with an additional IBP auxiliary objective results in large improvements, and increases IBP verification rates from 2.83% to 18.36% on SNLI, and from 7.77% to 17.44% on MNLI, compared to standard training.

As we have observed, only a minority of data points can be positively verified using IBP, or using exhaustive verification (where possible) – indicating a fundamental difficulty of learning to adhere to this specification, and generalising this behaviour to unseen test data. We thus conclude with a cautionary note on model undersensitivity: various defences are able to reduce the extent of the problem, but ridding models entirely of this undesirable behaviour comes at the cost of what models are developed for in the first place: generalisation to held-out test samples – at least when using the technical means considered here.

IBP is general enough to be extendable to other types of specifications, and one next step would be to adapt it to symbol substitutions, as explored in Chapter 6, rather than symbol deletions. Furthermore, the DAM model architecture explored here is amenable to bound propagation due to its limited depth and lack of context-dependent computation of word representations. We see overcoming these challenges as critical next steps to applying the IBP-based verification methodology to more recent generations of neural NLP models (Devlin et al., 2019).

Chapter 8

General Conclusion

8.1 Recapitulation

We have in this thesis investigated a series of problems in machine reading comprehension with a particular focus on dataset aspects. Following the introduction in Chapter 1 and general background in Chapter 2, we have considered three scenarios which are reflected in parts I-III of the thesis. In Part I / Chapter 3 we have modified the established crowdsourcing approach to RC dataset creation pioneered in SQUAD and used it to construct a multiple-choice dataset for the science exam QA domain. In Part II we have addressed the problem of creating datasets for cross-document multi-hop RC (Chapter 4), followed by an investigation of Pseudo-Relevance Feedback as the basis for selecting suitable combinations of documents for a model to process (Chapter 5). In Part III we have investigated the problem of RC model undersensitivity – a problem that is related to a lack of closely related unanswerable samples in RC training data (Chapter 6). Finally we have explored a formal verification approach to evaluate and address model undersensitivity using Interval Bound Propagation (Chapter 7). We will proceed with a summary of major contributions, major findings, and with critical reflections before concluding with perspectives on future research.

8.2 Major Contributions

New Dataset Resources: We have produced new RC dataset resources both for the domain of science exam QA (SCIQ), and for multi-hop RC across documents (WIKIHOP and MEDHOP). These datasets are publicly available and can serve as training resources and evaluation benchmarks for future RC research.

Innovation in Dataset Construction Methodology: The construction of these datasets was supported by innovations in dataset construction methodology. Concretely this includes a method to generate plausible multiple-choice answer candidates for science questions, and a graph traversal approach to construct multi-hop cross-document RC datasets using distant supervision.

Identification of Dataset Limitations: We have highlighted multiple pitfalls, dataset biases, and artefacts resulting from different data annotation strategies. This list of dataset biases includes i) a lack of plausible alternative answer candidates ii) label imbalance iii) spurious correlations between particular documents and answers in multi-document settings iv) a lack of structurally similar, yet unanswerable questions. We have then explored the following approaches to circumvent the listed issues, respectively: the inclusion of plausible alternative answer candidates and documents (i), sub-sampling (ii, iii), randomised masking (ii, iii), data augmentation and adversarial training (iv), as well as model verification (iv).

Retrieval for Document Combinations: We have shown that Pseudo-Relevance Feedback is a better suited retrieval strategy than TF-IDF or BM25 for selecting document combinations on WIKIHOP, and used this to improve both the document selection procedure and downstream accuracy of two neural RC models.

Model Undersensitivity: We have established and quantified the problem of RC model undersensitivity for natural language inputs, which had previously only been demonstrated for partially deleted inputs. After an examination of the phenomenon and characterisation of affected samples, we showed that it can be mitigated effectively through adversarial training and data augmentation, thus showing that the problem is largely attributable to a lack of structurally similar, but unanswerable training samples. We have furthermore shown that reducing a model's undersensitivity improves its behaviour when given a train / test distribution mismatch, and that it increases the model's robustness on the adversarial datasets from Jia and Liang (2017).

Formal Model Verification: We have introduced formal model verification as a method to evaluate a specification about an NLP model's undersensitivity. We have adapted Interval Bound Propagation to formally verify this specification for the DAM architecture (Parikh et al., 2016) in the NLI task in particular, evaluated and discussed its effectiveness and efficiency, and compared the verification and adversarial error rates of several training methods aimed at reducing undersensitivity.

8.3 Major Findings

Science Exam QA: The crowdsourcing approach to RC dataset creation can be extended to the science exam QA domain. Although questions are similar to real exam questions as judged by non-experts, they differ from them by being formulated using concrete documents, with which they show high levels of lexical overlap. We observed that both *Lucene* and *Aristo* (Clark et al., 2016) – two previously developed science exam solvers with access to a corpus of relevant documents, score higher on the resulting dataset compared to real exam questions. Neural RC methods achieve comparable performance to these prior science exam solvers when coupled with an IR component. Adding the newly assembled dataset as additional training data to existing exam questions improves the performance of two neural RC models evaluated on real exam questions.

Multi-Hop RC: We have demonstrated that it is possible to collect a noisy dataset of cross-document multi-hop RC samples using distant supervision and graph traversal in a bipartite graph of entities and documents. This dataset creation approach is however prone to biases, notably type consistency, label imbalance, and spurious correlations between particular documents and answers. These dataset biases can be mitigated by introducing alternative answer candidates and documents, via sub-sampling, and via entity masking. Without the application of these countermeasures statistical heuristics can achieve high accuracy values; even with the application of some of these mitigation measures they reach comparable performance to two neural RC models (*FastQA* and *BiDAF*). With the application of randomised entity masking in particular, the performance of shallow statistical heuristics drops, while neural RC models are able to retain their accuracy levels.

Methods for Selecting Document Combinations: The performance of neural RC methods on the collected multi-hop RC datasets improves when restricting the given documents to exclude irrelevant documents. This has inspired the sub-problem of identifying relevant document *combinations*, and we have shown that Pseudo-Relevance Feedback is better suited for this than TF-IDF or BM25, and obtained downstream accuracy improvements on WIKIHOP by exploiting this. These improvements were however not reflected in HOTPOTQA, which we related to differences in question length (and consequent retrieval vector sparsity), as well as differences in lexical overlap between query and documents.

Model Undersensitity: Commonly used neural RC methods trained on both SQUAD2.0 and NEWSQA are vulnerable to model undersensitivity attacks: adversarially chosen semantic changes to the comprehension question do not affect their answer prediction and even increase prediction confidence. We have identified a lack of structurally similar samples during training as responsible for this, and shown that if such samples are added, models become less vulnerable even to attacks based on new perturbations, become more robust to ADDSENT and ADDONESENT adversarial samples, and rely less on spurious type consistency cues to answer the question. Furthermore, undersensitivity attacks are more prevalent among samples with lower confidence and lower accuracy, and they transfer between ROBERTA and BERT models.

Undersensitivity Specification Verification: We have demonstrated that the DAM model does only for a small minority of samples verifiably adhere to an undersensitivity specification under word combination deletions. The rate of samples for which the verification is satisfied can be improved with the introduction of either adversarial, or IBP-related objectives: for example, verified accuracy can be increased from 2.8% to 18.4% on SNLI with an IBP-related objective. However, the modifications to conventional discriminative training that we tested come at the price of deteriorated nominal test accuracy. Besides this, we have shown that IBP-based verification is by orders of magnitude more efficient than exhaustive verification in the undersensitivity problem setting, while having overall low false negative ratios when evaluated in a setting where this evaluation is computationally feasible (e.g. 15.4% for a standard model on short sequences in SNLI).

8.4 Critical Reflection

Work in progress: A first – and important – point to emphasise is that the datasets we have constructed are not the final word on how we should train models to learn desirable RC behaviours. Instead, we see them as a step on the way towards improving our understanding of relevant factors in RC dataset construction and resulting model capabilities. The observations and perspectives we have described in this thesis can contribute to this understanding, and hopefully lead to new and further improved annotation methodologies and task conceptualisations.

Dataset limitations are not comprehensive: It is further worth mentioning that the dataset biases and limitations we have addressed – e.g. spurious co-occurrences of documents and answers in Chapter 4, or model undersensitivity in Chapter 6 – do not form a comprehensive list of such limitations. The diversity of different structural regularities in RC datasets suggests it to us as likely that additional biases and shortcuts will be discovered in the future.

Dataset noise: The dataset construction methods we have laid out do not always and reliably produce high-quality samples. While this is a general concern in data annotation, it is one for this work in particular, given our reliance on distant supervision and crowdsourced annotation by non-experts. We have highlighted this at various points and partly quantified the extent of this issue, e.g. by estimating the extent of distant supervision violations in Chapter 4.

Pre-trained models have changed the picture: The recent penetration of the NLP field with representations from pre-trained language models (Peters et al., 2018; Devlin et al., 2019) has shifted the picture regarding the role of RC training data: besides task-specific annotated samples, a large corpus of unlabeled pre-training data now also affects the resulting models. It is worth noting that models can capture factual information from their pre-training data (Petroni et al., 2019), which may enable them to predict correct answers to RC questions with less or without consideration of the text passage(s) given to them in an RC task. This introduces particular complications to the conceptual interpretation of multi-hop behaviour e.g. in WIKIHOP, which contains queries about paragraphs which are typically included in LM pre-training data (WIKIPEDIA). Models leveraging pre-trained representations blur the line between multi-hop and direct text comprehension, as it is unclear to what extent they rely on implicit background knowledge.

Limitation to the English language: All experiments in this thesis were conducted on English language corpora and datasets. We do not see any reason in principle for the transfer of most of our findings to other languages, yet some methodological contributions may have to be adapted depending on the language in consideration, e.g. the types of lexical permutations introduced in Part III on undersensitivity.

Limitations to applicability: The datasets we produced and the consequently trained models are not at a stage in which they are ready for direct practical application. Instead, they can be considered a testbed which contributes to an understanding of potentially relevant factors for practical applications in the future.

8.5 Future Work

Identification of dataset and model limitations: As RC progresses through the collective efforts of the research community, pinpointing and systematising particular RC failure modes will continue to play an important role in improving models, and likely become more conceptually challenging with increasing model capabilities. Having largely traded away model interpretability for generalisation performance in neural RC, new methods that help us understand model behaviours – especially undesirable ones – are necessary to further improve RC systems and datasets. We have in this thesis focused mostly on data-related aspects, but there are many ways in which undesirable model behaviour can be identified. Model diagnostics (Ribeiro et al., 2016, 2018a), crowdsourcing the detection of RC model failures (Bartolo et al., 2020), or adopting adversarial perspectives (Jia and Liang, 2017) are promising directions to identify and systematise model limitations; the latter can ideally lead to an ongoing virtuous cycle of new adversarial attacks and responses. With an improved understanding of dataset and model shortcomings we can test countermeasures and develop technical solutions that gradually close the gap between machine and human reading comprehension ability.

Reconsideration of dataset needs for science exam QA: In part I of this thesis we have considered a particular domain – science exam QA. Since the time of this study, neural models have substantially progressed (Clark et al., 2019). A thorough reconsideration of dataset requirements, in particular for the effective use of models relying on pre-trained LM representations would both be insightful and useful to further improve these models and lift them to more challenging benchmarks.

Future work on multi-hop datasets: A number of additional datasets targeting multi-hop RC has been assembled besides WIKIHOP and MEDHOP, using different annotation paradigms (Talmor and Berant, 2018; Yang et al., 2018; Khashabi et al., 2018a). Subsequent work has also shown limitations to the degree of multi-hop inference necessary, e.g. in WIKIHOP and HOTPOTQA (Jiang and Bansal, 2019; Min et al., 2019; Chen and Durrett, 2019). Future dataset construction efforts towards multi-hop comprehension should take into accounts these lessons and conceive annotation methodologies – potentially using data augmentation strategies – to circumvent the ability of models to learn such shortcuts.

Further explorations into model undersensitivity: The undersensitivity problem we considered would benefit from further thorough investigation and analysis, e.g. with the use of different perturbation spaces and countermeasures, and a better theoretical framework to capture the phenomenon.

Neural network verification in NLP: The formal verification of particular properties for neural NLP models is still in a nascent stage. We see promise in this conceptual framework, as it allows us to retain guarantees and a degree of control over otherwise freely optimised neural structures – *if* we are able to precisely define the model behaviour we desire with a formal specification. Future work in this direction includes improving the looseness of verification bounds, the adaptation of verification approaches to larger and more potent model architectures in NLP, and conceptual work to formally define specifications that address desirable and undesirable model behaviours which go beyond fitting correct labels to individual data points.

Appendix A

SciQ: List of Study Books

The following is a list of the books we used as data source:

- OpenStax, Anatomy & Physiology. OpenStax. 25 April 2013[1]
- OpenStax, Biology. OpenStax. May 20, 2013[2]
- OpenStax, Chemistry. OpenStax. 11 March 2015[3]
- OpenStax, College Physics. OpenStax. 21 June 2012[4]
- OpenStax, Concepts of Biology. OpenStax. 25 April 2013[5]
- Biofundamentals 2.0 – by Michael Klymkowsky, University of Colorado & Melanie Cooper, Michigan State University[6]
- Earth Systems, An Earth Science Course on www.curriki.org[7]
- General Chemistry, Principles, Patterns, and Applications by Bruce Averill, Strategic Energy Security Solutions and Patricia Eldredge, R.H. Hand, LLC; Saylor Foundation[8]

[1] Download for free at http://cnx.org/content/col11496/latest/

[2] Download for free at http://cnx.org/content/col11448/latest/

[3] Download for free at http://cnx.org/content/col11760/latest/

[4] Download for free at http://cnx.org/content/col11406/latest

[5] Download for free at http://cnx.org/content/col11487/latest

[6] https://open.umn.edu/opentextbooks/BookDetail.aspx?bookId=350

[7] http://www.curriki.org/xwiki/bin/view/Group_CLRN-OpenSourceEarthScienceCourse/

[8] https://www.saylor.org/site/textbooks/General%20Chemistry%20Principles,%20Patterns,%20and%20Applications.pdf

- General Biology; Paul Doerder, Cleveland State University & Ralph Gibson, Cleveland State University [9]

- Introductory Chemistry by David W. Ball, Cleveland State University. Saylor Foundation [10]

- The Basics of General, Organic, and Biological Chemistry by David Ball, Cleveland State University & John Hill, University of Wisconsin & Rhonda Scott, Southern Adventist University. Saylor Foundation[11]

- Barron's New York State Grade 4 Elementary-Level Science Test, by Joyce Thornton Barry and Kathleen Cahill [12]

- Campbell Biology: Concepts & Connections by Jane B. Reece, Martha R. Taylor, Eric J. Simon, Jean L. Dickey[13]

- CK-12 Peoples Physics Book Basic [14]

- CK-12 Biology Advanced Concepts [15]

- CK-12 Biology Concepts [16]

- CK-12 Biology [17]

- CK-12 Chemistry - Basic [18]

- CK-12 Chemistry Concepts – Intermediate [19]

[9] https://upload.wikimedia.org/wikipedia/commons/4/40/GeneralBiology.pdf

[10] https://www.saylor.org/site/textbooks/Introductory%20Chemistry.pdf

[11] http://web.archive.org/web/20131024125808/http://www.saylor.org/site/textbooks/The%20Basics%20of%20General,%20Organic%20and%20Biological%20Chemistry.pdf

[12] We do not include documents from this resource in the shared dataset.

[13] We do not include documents from this resource in the shared dataset.

[14] http://www.ck12.org/book/Peoples-Physics-Book-Basic/

[15] http://www.ck12.org/book/CK-12-Biology-Advanced-Concepts/

[16] http://www.ck12.org/book/CK-12-Biology-Concepts/

[17] http://www.ck12.org/book/CK-12-Biology/

[18] http://www.ck12.org/book/CK-12-Chemistry-Basic/

[19] http://www.ck12.org/book/CK-12-Chemistry-Concepts-Intermediate/

- CK-12 Earth Science Concepts For Middle School[20]
- CK-12 Earth Science Concepts For High School[21]
- CK-12 Earth Science For Middle School [22]
- CK-12 Life Science Concepts For Middle School [23]
- CK-12 Life Science For Middle School [24]
- CK-12 Physical Science Concepts For Middle School[25]
- CK-12 Physical Science For Middle School [26]
- CK-12 Physics Concepts - Intermediate [27]
- CK-12 People's Physics Concepts [28]

CK-12 books were obtained under the Creative Commons Attribution-Non-Commercial 3.0 Unported (CC BY-NC 3.0) License. [29]

[20] http://www.ck12.org/book/CK-12-Earth-Science-Concepts-For-Middle-School/
[21] http://www.ck12.org/book/CK-12-Earth-Science-Concepts-For-High-School/
[22] http://www.ck12.org/book/CK-12-Earth-Science-For-Middle-School/
[23] http://www.ck12.org/book/CK-12-Life-Science-Concepts-For-Middle-School/
[24] http://www.ck12.org/book/CK-12-Life-Science-For-Middle-School/
[25] http://www.ck12.org/book/CK-12-Physical-Science-Concepts-For-Middle-School/
[26] http://www.ck12.org/book/CK-12-Physical-Science-For-Middle-School/
[27] http://www.ck12.org/book/CK-12-Physics-Concepts-Intermediate/
[28] http://www.ck12.org/book/Peoples-Physics-Concepts/
[29] http://creativecommons.org/licenses/by-nc/3.0/

Bibliography

M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

M. Adams and A. Collins. A schema-theoretic view of reading. technical report no. 32. *Bolt Beranek and Newman Inc.*, 1977.

M. Agarwal and P. Mannem. Automatic gap-fill question generation from text books. In *Proceedings of the 6th Workshop on Innovative Use of NLP for Building Educational Applications*, IUNLPBEA '11, pages 56–64, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 9781937284039. URL `http://dl.acm.org/citation.cfm?id=2043132.2043139`.

E. Agirre and P. Edmonds. *Word Sense Disambiguation: Algorithms and Applications*. Springer Publishing Company, Incorporated, 1st edition, 2007. ISBN 1402068700.

E. Agirre and D. Martinez. Knowledge sources for word sense disambiguation. In *Proceedings of the 4th International Conference on Text, Speech and Dialogue*, TSD '01, page 1–10, Berlin, Heidelberg, 2001. Springer-Verlag. ISBN 3540425578.

C. Alberti, D. Andor, E. Pitler, J. Devlin, and M. Collins. Synthetic QA corpora generation with roundtrip consistency. In *ACL (1)*, pages 6168–6173. Association for Computational Linguistics, 2019.

I. Aldabe and M. Maritxalar. *Automatic Distractor Generation for Domain Specific Texts*, pages 27–38. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. ISBN 978-3-642-14770-8.

M. Alzantot, Y. Sharma, A. Elgohary, B.-J. Ho, M. Srivastava, and K.-W. Chang. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1316. URL `https://www.aclweb.org/anthology/D18-1316`.

R. C. Anderson, R. E. Reynolds, D. L. Schallert, and E. T. Goetz. Frameworks for comprehending discourse. *American Educational Research Journal*, 14(4): 367–381, 1977. doi: 10.3102/00028312014004367. URL `https://doi.org/10.3102/00028312014004367`.

A. Athalye, N. Carlini, and D. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018.

H. K. Azad and A. Deepak. Query expansion techniques for information retrieval: a survey. *CoRR*, abs/1708.00247, 2017.

A. Bairoch, B. Boeckmann, S. Ferro, and E. Gasteiger. Swiss-Prot: Juggling between evolution and stability. *Briefings in Bioinformatics*, 5(1):39–55, 2004. doi: 10.1093/bib/5.1.39.

C. F. Baker, C. J. Fillmore, and J. B. Lowe. The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Lin-*

guistics - Volume 1, ACL ’98/COLING ’98, page 86–90, USA, 1998. Association for Computational Linguistics. doi: 10.3115/980845.980860. URL `https://doi.org/10.3115/980845.980860`.

M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction from the web. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI’07, pages 2670–2676, 2007. URL `http://dl.acm.org/citation.cfm?id=1625275.1625705`.

V. Barr and J. L. Klavans. Verification and validation of language processing systems: Is it evaluation? In *Proceedings of the ACL 2001 Workshop on Evaluation Methodologies for Language and Dialogue Systems*, 2001. URL `https://www.aclweb.org/anthology/W01-0906`.

F. C. Bartlett. Remembering: A study in experimental and social psychology. *Cambridge University Press*, 1932.

M. Bartolo, A. Roberts, J. Welbl, S. Riedel, and P. Stenetorp. Beat the ai: Investigating adversarial human annotations for reading comprehension, 2020.

Y. Belinkov and Y. Bisk. Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=BJ8vJebC-`.

J. Berant, A. Chou, R. Frostig, and P. Liang. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, 2013. URL `http://aclweb.org/anthology/D/D13/D13-1160.pdf`.

B. S. Bloom, M. B. Engelhart, E. J. Furst, W. H. Hill, and D. R. Krathwohl. *Taxonomy of educational objectives. The classification of educational goals. Handbook 1: Cognitive domain.* Longmans Green, New York, 1956.

K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIG-*

MOD 08 Proceedings of the 2008 ACM SIGMOD international conference on Management of data, pages 1247–1250, 2008. ISBN 9781605581026. doi: 10.1145/1376616.1376746.

A. Bordes, S. Chopra, and J. Weston. Question answering with subgraph embeddings. In *Empirical Methods for Natural Language Processing (EMNLP)*, pages 615–620, 2014.

A. Bordes, N. Usunier, S. Chopra, and J. Weston. Large-scale simple question answering with memory networks. *CoRR*, abs/1506.02075, 2015. URL `http://arxiv.org/abs/1506.02075`.

S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642. Association for Computational Linguistics, 2015. doi: 10.18653/v1/D15-1075. URL `http://aclweb.org/anthology/D15-1075`.

J. Boyd-Graber, B. Satinoff, H. He, and H. Daumé, III. Besting the quiz master: Crowdsourcing incremental classification games. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 1290–1301, 2012. URL `http://dl.acm.org/citation.cfm?id=2390948.2391094`.

L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, 1984. ISBN 0-534-98053-8.

C. Buck, J. Bulian, M. Ciaramita, A. Gesmundo, N. Houlsby, W. Gajewski, and W. Wang. Ask the right questions: Active question reformulation with reinforcement learning. *International Conference on Learning Representations (ICLR)*, 2018.

R. Bunel, I. Turkaslan, P. H. Torr, P. Kohli, and M. P. Kumar. Piecewise linear neural network verification: a comparative study. *arXiv preprint arXiv:1711.00455*, 2017.

R. Bunescu and R. Mooney. Learning to extract relations from the web using minimal supervision. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 576–583, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/P07-1073`.

C. Cardie. Empirical methods in information extraction. *AI magazine*, pages 65–79, 1997.

N. Carlini and D. Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 3–14. ACM, 2017.

A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka, and T. M. Mitchell. Toward an architecture for never-ending language learning. In *In AAAI*, 2010.

L. Carlson, D. Marcu, and M. E. Okurovsky. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue*, 2001. URL `https://www.aclweb.org/anthology/W01-1605`.

C. Carpineto and G. Romano. A survey of automatic query expansion in information retrieval. *ACM Comput. Surv.*, 44(1):1:1–1:50, Jan. 2012. ISSN 0360-0300. doi: 10.1145/2071389.2071390.

E. Charniak. Organization and inference in a frame-like system of common sense knowledge. In *Theoretical Issues in Natural Language Processing*, 1975. URL `https://www.aclweb.org/anthology/T75-2010`.

D. Chen, J. Bolton, and C. D. Manning. 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, 2016. URL `http://www.aclweb.org/anthology/P16-1223`.

D. Chen, A. Fisch, J. Weston, and A. Bordes. 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. Association for Computational Linguistics, 2017a. URL `http://www.aclweb.org/anthology/P17-1171`.

J. Chen and G. Durrett. Understanding dataset design choices for multi-hop reasoning. 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 4026–4032, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1405. URL `https://www.aclweb.org/anthology/N19-1405`.

Q. Chen, X. Zhu, Z.-H. Ling, S. Wei, H. Jiang, and D. Inkpen. Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668. Association for Computational Linguistics, 2017b. doi: 10.18653/v1/P17-1152. URL `http://aclweb.org/anthology/P17-1152`.

W. Chen, W. Xiong, X. Yan, and W. Y. Wang. Variational knowledge graph reasoning. In M. A. Walker et al., editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018*, pages 1823–1832. Association for Computational Linguistics, 2018.

C.-H. Cheng, G. Nührenberg, and H. Ruess. Maximum resilience of artificial neural networks. In *International Symposium on Automated Technology for Verification and Analysis*, pages 251–268. Springer, 2017.

K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. 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, Oct. 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1179. URL https://www.aclweb.org/anthology/D14-1179.

C. Clark and M. Gardner. 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. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/P18-1078.

P. Clark. Elementary school science and math tests as a driver for ai: Take the aristo challenge! In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 4019–4021. AAAI Press, 2015. ISBN 0-262-51129-0. URL http://dl.acm.org/citation.cfm?id=2888116.2888274.

P. Clark, P. Harrison, and N. Balasubramanian. A study of the knowledge base requirements for passing an elementary science test. In *Proceedings of the 2013 Workshop on Automated Knowledge Base Construction*, AKBC '13, pages 37–42, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2411-3. doi: 10.1145/2509558.2509565. URL http://doi.acm.org/10.1145/2509558.2509565.

P. Clark, O. Etzioni, T. Khot, A. Sabharwal, O. Tafjord, P. Turney, and D. Khashabi. Combining retrieval, statistics, and inference to answer elementary science questions. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 2580–2586. AAAI Press, 2016.

P. Clark, O. Etzioni, T. Khot, B. D. Mishra, K. Richardson, A. Sabharwal, C. Schoenick, O. Tafjord, N. Tandon, S. Bhakthavatsalam, et al. From 'F'to 'A'on the NY regents science exams: An overview of the aristo project. *arXiv preprint arXiv:1909.01958*, 2019.

R. Correia, J. Baptista, N. Mamede, I. Trancoso, and M. Eskenazi. Automatic generation of cloze question distractors. In *Proceedings of the Interspeech 2010*

Satellite Workshop on Second Language Studies: Acquisition, Learning, Education and Technology, Waseda University, Tokyo, Japan, 2010.

G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS)*, pages 303–314, 1989.

I. Dagan, O. Glickman, and B. Magnini. The pascal recognising textual entailment challenge. In *Proceedings of the First International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment*, MLCW'05, pages 177–190, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 3-540-33427-0, 978-3-540-33427-9. doi: 10.1007/11736790_9. URL `http://dx.doi.org/10.1007/11736790_9`.

R. Das, A. Neelakantan, D. Belanger, and A. McCallum. Chains of reasoning over entities, relations, and text using recurrent neural networks. *European Chapter of the Association for Computational Linguistics (EACL)*, pages 132–141, 2017.

D. Davidson. The logical form of action sentences. In N. Rescher, editor, *The Logic of Decision and Action*, pages 81–120. Univ. of Pittsburgh Press, 1967.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. 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, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL `https://www.aclweb.org/anthology/N19-1423`.

B. Dhingra, H. Liu, W. W. Cohen, and R. Salakhutdinov. Gated-attention readers for text comprehension. *CoRR*, abs/1606.01549, 2016. URL `http://arxiv.org/abs/1606.01549`.

B. Dhingra, K. Mazaitis, and W. W. Cohen. Quasar: Datasets for question answering by search and reading. *CoRR*, abs/1707.03904, 2017.

D. Dua, Y. Wang, P. Dasigi, G. Stanovsky, S. Singh, and M. Gardner. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. 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 2368–2378, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1246. URL `https://www.aclweb.org/anthology/N19-1246`.

M. Dunn, L. Sagun, M. Higgins, V. U. Güney, V. Cirik, and K. Cho. SearchQA: A new Q&A dataset augmented with context from a search engine. *CoRR*, abs/1704.05179, 2017. URL `http://arxiv.org/abs/1704.05179`.

K. Dvijotham, S. Gowal, R. Stanforth, R. Arandjelovic, B. O'Donoghue, J. Uesato, and P. Kohli. Training verified learners with learned verifiers. *arXiv preprint arXiv:1805.10265*, 2018a.

K. Dvijotham, R. Stanforth, S. Gowal, T. Mann, and P. Kohli. A dual approach to scalable verification of deep networks. *arXiv preprint arXiv:1803.06567*, 2018b.

J. Ebrahimi, A. Rao, D. Lowd, and D. Dou. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2006. URL `https://www.aclweb.org/anthology/P18-2006`.

A. Ettinger, S. Rao, H. Daumé III, and E. M. Bender. Towards linguistically generalizable NLP systems: A workshop and shared task. In *Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems*, 2017.

A. Fabregat, K. Sidiropoulos, P. Garapati, M. Gillespie, K. Hausmann, R. Haw, B. Jassal, S. Jupe, F. Korninger, S. McKay, L. Matthews, B. May, M. Milacic, K. Rothfels, V. Shamovsky, M. Webber, J. Weiser, M. Williams, G. Wu, L. Stein,

H. Hermjakob, and P. D'Eustachio. The Reactome pathway knowledgebase. *Nucleic Acids Research*, 44(D1):D481–D487, 2016. doi: 10.1093/nar/gkv1351.

S. Feng, E. Wallace, A. Grissom II, M. Iyyer, P. Rodriguez, and J. Boyd-Graber. Pathologies of neural models make interpretations difficult. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3719–3728. Association for Computational Linguistics, 2018. URL `http://aclweb.org/anthology/D18-1407`.

C. Fillmore. Frame semantics. In *Linguistics in the Morning Calm, ed. by The Linguistic Society of Korea*, pages 111–137, Soeul: Hanshin, 1982. The Linguistic Society of Korea.

J. R. Firth. A synopsis of linguistic theory 1930-55. *Oxford University Press*, 1957.

D. Fried, P. Jansen, G. Hahn-Powell, M. Surdeanu, and P. Clark. Higher-order lexical semantic models for non-factoid answer reranking. *Transactions of the Association of Computational Linguistics*, 3:197–210, 2015. URL `http://aclanthology.coli.uni-saarland.de/pdf/Q/Q15/Q15-1015.pdf`.

M. Gardner, P. P. Talukdar, B. Kisiel, and T. M. Mitchell. Improving learning and inference in a large knowledge-base using latent syntactic cues. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 833–838, 2013.

L. Getoor and B. Taskar. *Introduction to statistical relational learning*. The MIT Press, 2007.

M. Geva and J. Berant. Learning to search in long documents using document structure. In E. M. Bender et al., editors, *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018*, pages 161–176. Association for Computational Linguistics, 2018.

M. Glockner, V. Shwartz, and Y. Goldberg. Breaking NLI systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meet-*

ing of the Association for Computational Linguistics (Volume 2: Short Papers), pages 650–655, Melbourne, Australia, July 2018. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/P18-2103`.

S. Gowal, K. Dvijotham, R. Stanforth, R. Bunel, C. Qin, J. Uesato, R. Arandjelovic, T. A. Mann, and P. Kohli. On the effectiveness of interval bound propagation for training verifiably robust models. *CoRR*, abs/1810.12715, 2018. URL `http://arxiv.org/abs/1810.12715`.

A. Graves. Adaptive computation time for recurrent neural networks. *CoRR*, abs/1603.08983, 2016. URL `http://arxiv.org/abs/1603.08983`.

H. Gurulingappa, A. M. Rajput, A. Roberts, J. Fluck, M. Hofmann-Apitius, and L. Toldo. Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. *Journal of Biomedical Informatics*, 45(5):885 – 892, 2012. ISSN 1532-0464. doi: http://dx.doi.org/10.1016/j.jbi.2012.04.008. Text Mining and Natural Language Processing in Pharmacogenomics.

S. Gururangan, S. Swayamdipta, O. Levy, R. Schwartz, S. Bowman, and N. A. Smith. Annotation artifacts in natural language inference data. 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 107–112, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2017. URL `https://www.aclweb.org/anthology/N18-2017`.

Z. S. Harris. Distributional structure. *Routledge*, 10(2-3):146–162, 1954. doi: 10.1080/00437956.1954.11659520.

M. Heilman and N. A. Smith. 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, 2010. Association for

Computational Linguistics. ISBN 1-932432-65-5. URL `http://dl.acm.org/citation.cfm?id=1857999.1858085`.

K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom. 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., 2015. URL `http://papers.nips.cc/paper/5945-teaching-machines-to-read-and-comprehend.pdf`.

D. Hewlett, A. Lacoste, L. Jones, I. Polosukhin, A. Fandrianto, J. Han, M. Kelcey, and D. Berthelot. WikiReading: A novel large-scale language understanding task over Wikipedia. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1545, Berlin, Germany, Aug. 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1145. URL `https://www.aclweb.org/anthology/P16-1145`.

F. Hill, A. Bordes, S. Chopra, and J. Weston. The goldilocks principle: Reading children's books with explicit memory representations. *ICLR*, 2016.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9 (8):1735–1780, 1997.

P. M. Htut, S. R. Bowman, and K. Cho. Training a ranking function for open-domain question answering. In S. R. Cordeiro et al., editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 2-4, 2018, Student Research Workshop*, pages 120–127. Association for Computational Linguistics, 2018.

M. Hu, Y. Peng, and X. Qiu. Mnemonic reader for machine comprehension. *CoRR*, abs/1705.02798, 2017. URL `http://arxiv.org/abs/1705.02798`.

M. Hu, F. Wei, Y. Peng, Z. Huang, N. Yang, and D. Li. Read + verify: Machine reading comprehension with unanswerable questions. In *AAAI*, pages 6529–6537. AAAI Press, 2019.

P.-S. Huang, R. Stanforth, J. Welbl, C. Dyer, D. Yogatama, S. Gowal, K. Dvijotham, and P. Kohli. Achieving verified robustness to symbol substitutions via interval bound propagation. *arXiv preprint arXiv:1909.01492*, 2019.

A. Iran-Nejad and A. Winsler. Bartlett's schema theory and modern accounts of learning and remembering. *Journal of Mind and Behavior*, 21:5–35, 10 2000.

M. Iyyer, J. Wieting, K. Gimpel, and L. Zettlemoyer. Adversarial example generation with syntactically controlled paraphrase networks. 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 1875–1885, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1170. URL `https://www.aclweb.org/anthology/N18-1170`.

J.-H. Jacobsen, J. Behrmann, R. Zemel, and M. Bethge. Excessive invariance causes adversarial vulnerability. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=BkfbpsAcF7`.

S. Jain. Question answering over knowledge base using factual memory networks. In *Proceedings of NAACL-HLT*, pages 109–115, 2016.

P. Jansen, N. Balasubramanian, M. Surdeanu, and P. Clark. What's in an explanation? characterizing knowledge and inference requirements for elementary science exams. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2956–2965, Osaka, Japan, Dec. 2016. The COLING 2016 Organizing Committee. URL `https://www.aclweb.org/anthology/C16-1278`.

P. Jansen, R. Sharp, M. Surdeanu, and P. Clark. Framing QA as building and ranking intersentence answer justifications. *Computational Linguistics*, 43(2):407–

449, 2017. doi: 10.1162/COLI_a_00287. URL `https://doi.org/10.1162/COLI_a_00287`.

R. Jia and P. Liang. 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, Sept. 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1215. URL `https://www.aclweb.org/anthology/D17-1215`.

R. Jia, A. Raghunathan, K. Göksel, and P. Liang. Certified robustness to adversarial word substitutions. *arXiv preprint arXiv:1909.00986*, 2019.

Y. Jiang and M. Bansal. Avoiding reasoning shortcuts: Adversarial evaluation, training, and model development for multi-hop QA. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2726–2736, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1262. URL `https://www.aclweb.org/anthology/P19-1262`.

Z. Jiang, F. F. Xu, J. Araki, and G. Neubig. How can we know what language models know?, 2019.

M. Joshi, E. Choi, D. S. Weld, and L. Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, July 2017.

A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431. Association for Computational Linguistics, 2017. URL `http://aclweb.org/anthology/E17-2068`.

R. Kadlec, M. Schmid, O. Bajgar, and J. Kleindienst. Text understanding with the attention sum reader network. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 908—-918, 2016.

N. Kalchbrenner, E. Grefenstette, and P. Blunsom. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665, Baltimore, Maryland, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/P14-1062. URL `https://www.aclweb.org/anthology/P14-1062`.

D. Kang, T. Khot, A. Sabharwal, and E. Hovy. AdvEntuRe: Adversarial training for textual entailment with knowledge-guided examples. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2418–2428, Melbourne, Australia, July 2018. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/P18-1225`.

I. Kant. *Critique of pure reason, trans*. Cambridge University Press, 1781.

G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*, pages 97–117. Springer, 2017.

D. Khashabi, T. Khot, A. Sabharwal, P. Clark, O. Etzioni, and D. Roth. Question answering via integer programming over semi-structured knowledge. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 1145–1152, 2016. URL `http://www.ijcai.org/Abstract/16/166`.

D. Khashabi, S. Chaturvedi, M. Roth, S. Upadhyay, and D. Roth. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. 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 252–262, New Orleans, Louisiana, June 2018a. Association for Computational Linguistics. doi: 10.18653/v1/N18-1023. URL `https://www.aclweb.org/anthology/N18-1023`.

D. Khashabi, T. Khot, A. Sabharwal, and D. Roth. Question answering as global reasoning over semantic abstractions. In S. A. McIlraith et al., editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18)*, pages 1905–1914. AAAI Press, 2018b.

T. Khot, N. Balasubramanian, E. Gribkoff, A. Sabharwal, P. Clark, and O. Etzioni. Exploring markov logic networks for question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 685–694, 2015. URL `http://aclweb.org/anthology/D/D15/D15-1080.pdf`.

Y. Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, Oct. 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1181. URL `https://www.aclweb.org/anthology/D14-1181`.

T. Kočiský, J. Schwarz, P. Blunsom, C. Dyer, K. M. Hermann, G. Melis, and E. Grefenstette. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328, 2018. doi: 10.1162/tacl_a_00023. URL `https://www.aclweb.org/anthology/Q18-1023`.

D. R. Krathwohl. A revision of Bloom's Taxonomy: an overview – Benjamin S. Bloom, University of Chicago. *Theory Into Practice*, 42(4), Autumn 2002. `http://www.findarticles.com/p/articles/mi_m0NQM/is_4_41/ai_94872707/print` – last visited 12th May 2008.

A. Kumar, O. Irsoy, P. Ondruska, M. Iyyer, I. G. James Bradbury, V. Zhong, R. Paulus, and R. Socher. Ask me anything: Dynamic memory networks for natural language processing. *International Conference on Machine Learning*, 48:1378–1387, 2016.

A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, M. Kelcey, J. Devlin, K. Lee, K. N. Toutanova, L. Jones, M.-W. Chang, A. Dai, J. Uszkoreit, Q. Le, and S. Petrov. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*, 2019.

G. Lai, Q. Xie, H. Liu, Y. Yang, and E. Hovy. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1082. URL `https://www.aclweb.org/anthology/D17-1082`.

J. R. Landis and G. G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, pages 159–174, 1977.

N. Lao and W. W. Cohen. Relational retrieval using a combination of path-constrained random walks. *Machine learning*, 81(1):53–67, 2010.

N. Lao, T. Mitchell, and W. W. Cohen. Random walk inference and learning in a large scale knowledge base. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 529–539, 2011.

V. Law, C. Knox, Y. Djoumbou, T. Jewison, A. C. Guo, Y. Liu, A. Maciejewski, D. Arndt, M. Wilson, V. Neveu, A. Tang, G. Gabriel, C. Ly, S. Adamjee, Z. T. Dame, B. Han, Y. Zhou, and D. S. Wishart. DrugBank 4.0: Shedding new light on drug metabolism. *Nucleic Acids Research*, 42(D1):D1091–D1097, 2014. doi: 10.1093/nar/gkt1068.

O. Levy and Y. Goldberg. Neural word embedding as implicit matrix factorization. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2177–2185. Curran Associates, Inc., 2014. URL `http://papers.nips.cc/paper/`

5477-neural-word-embedding-as-implicit-matrix-factorization.pdf.

O. Levy, M. Seo, E. Choi, and L. Zettlemoyer. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342, August 2017.

M. Lewis and A. Fan. Generative question answering: Learning to answer the whole question. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkx0RjA9tX.

Y. Li and P. Clark. Answering elementary science questions by constructing coherent scenes using background knowledge. In *EMNLP*, pages 2007–2012, 2015.

Y. Li, T. Cohn, and T. Baldwin. Robust training under linguistic adversity. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 21–27, 2017.

D. Lin and P. Pantel. Discovery of inference rules for question-answering. *Nat. Lang. Eng.*, 7(4):343–360, Dec. 2001. ISSN 1351-3249. doi: 10.1017/S1351324901002765. URL http://dx.doi.org/10.1017/S1351324901002765.

F. Liu and J. Perez. Gated end-to-end memory networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Volume 1: Long Papers*, pages 1–10, 2017. URL http://aclanthology.info/papers/E17-1001/gated-end-to-end-memory-networks.

Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019. URL http://arxiv.org/abs/1907.11692.

M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993. URL `https://www.aclweb.org/anthology/J93-2004`.

Mausam, M. Schmitz, R. Bart, S. Soderland, and O. Etzioni. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 523–534, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=2390948.2391009`.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013a.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013b.

G. A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11): 39–41, Nov. 1995. ISSN 0001-0782. doi: 10.1145/219717.219748. URL `http://doi.acm.org/10.1145/219717.219748`.

S. Min, V. Zhong, R. Socher, and C. Xiong. 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, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1160. URL `https://www.aclweb.org/anthology/P18-1160`.

S. Min, E. Wallace, S. Singh, M. Gardner, H. Hajishirzi, and L. Zettlemoyer. Compositional questions do not necessitate multi-hop reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguis-*

tics, pages 4249–4257, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1416. URL `https://www.aclweb.org/anthology/P19-1416`.

P. Minervini and S. Riedel. Adversarially regularising neural nli models to integrate logical background knowledge. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 65–74. Association for Computational Linguistics, 2018. URL `http://aclweb.org/anthology/K18-1007`.

M. Minsky. A framework for representing knowledge. *MIT Artificial Intelligence Laboratory, Memo 306*, 1974.

M. Mintz, S. Bills, R. Snow, and D. Jurafsky. 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, 2009. URL `http://www.aclweb.org/anthology/P/P09/P09-1113`.

M. Mirman, T. Gehr, and M. Vechev. Differentiable abstract interpretation for provably robust neural networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 3578–3586, 2018.

R. Mitkov and L. A. Ha. Computer-aided generation of multiple-choice tests. In *Proceedings of the HLT-NAACL 03 Workshop on Building Educational Applications Using Natural Language Processing - Volume 2*, HLT-NAACL-EDUC ’03, pages 17–22, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1118894.1118897. URL `http://dx.doi.org/10.3115/1118894.1118897`.

R. Mitkov, L. A. Ha, A. Varga, and L. Rello. Semantic similarity of distractors in multiple-choice tests: Extrinsic evaluation. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, GEMS ’09, pages 49–56, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=1705415.1705422`.

A. Morales, V. Premtoon, C. Avery, S. Felshin, and B. Katz. Learning to answer questions from Wikipedia infoboxes. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1930–1935, 2016. URL `https://aclweb.org/anthology/D16-1199`.

J. Mostow and H. Jang. Generating diagnostic multiple choice comprehension cloze questions. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 136–146, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=2390384.2390401`.

P. K. Mudrakarta, A. Taly, M. Sundararajan, and K. Dhamdhere. Did the model understand the question? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1896–1906, Melbourne, Australia, July 2018. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/P18-1176`.

R. Musa, X. Wang, A. Fokoue, N. Mattei, M. Chang, P. Kapanipathi, B. Makni, K. Talamadupula, and M. Witbrock. Answering science exam questions using query rewriting with background knowledge. *CoRR*, abs/1809.05726, 2018.

D. Nadeau and S. Sekine. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26, January 2007. URL `http://www.ingentaconnect.com/content/jbp/li/2007/00000030/00000001/art00002`. Publisher: John Benjamins Publishing Company.

K. Narasimhan, A. Yala, and R. Barzilay. Improving information extraction by acquiring external evidence with reinforcement learning. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2355–2365, Austin, Texas, Nov. 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1261. URL `https://www.aclweb.org/anthology/D16-1261`.

A. Neelakantan, B. Roth, and A. McCallum. Compositional vector space models for knowledge base completion. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 156—166, 2015.

M. Neumann, P. Stenetorp, and S. Riedel. Learning to reason with adaptive computation. In *Interpretable Machine Learning for Complex Systems at the 2016 Conference on Neural Information Processing Systems (NIPS)*, Barcelona, Spain, December 2016.

T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, and L. Deng. MS MARCO: A human generated machine reading comprehension dataset. *CoRR*, abs/1611.09268, 2016. URL `http://arxiv.org/abs/1611.09268`.

K. Nishida, I. Saito, A. Otsuka, H. Asano, and J. Tomita. Retrieve-and-read: Multi-task learning of information retrieval and reading comprehension. In A. Cuzzocrea et al., editors, *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018*, pages 647–656. ACM, 2018.

T. Niu and M. Bansal. Adversarial over-sensitivity and over-stability strategies for dialogue models. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 486–496, Brussels, Belgium, Oct. 2018. Association for Computational Linguistics. doi: 10.18653/v1/K18-1047. URL `https://www.aclweb.org/anthology/K18-1047`.

R. Nogueira and K. Cho. WebNav: A new large-scale task for natural language based sequential decision making. *CoRR*, abs/1602.02261, 2016.

R. Nogueira and K. Cho. Task-oriented query reformulation with reinforcement learning. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 574—-583, 2017.

T. Onishi, H. Wang, M. Bansal, K. Gimpel, and D. A. McAllester. Who did what: A large-scale person-centered cloze dataset. In *Proceedings of the 2016 Conference* *on Empirical Methods in Natural Language Processing, EMNLP 2016*, pages 2230–2235, 2016. URL `http://aclweb.org/anthology/D/D16/D16-1241.pdf`.

M. Palmer, D. Gildea, and N. Xue. *Semantic Role Labeling*. Morgan and Claypool Publishers, 1st edition, 2010. ISBN 1598298313.

A. Papasalouros, K. Kanaris, and K. Kotis. Automatic generation of multiple choice questions from domain ontologies. In M. B. Nunes and M. McPherson, editors, *e-Learning*, pages 427–434. IADIS, 2008. ISBN 978-972-8924-58-4.

D. Paperno, G. Kruszewski, A. Lazaridou, N. Q. Pham, R. Bernardi, S. Pezzelle, M. Baroni, G. Boleda, and R. Fernandez. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1525–1534, 2016. doi: 10.18653/v1/P16-1144. URL `http://www.aclweb.org/anthology/P16-1144`.

A. Parikh, O. Täckström, D. Das, and J. Uszkoreit. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255. Association for Computational Linguistics, 2016. doi: 10.18653/v1/D16-1244. URL `http://aclweb.org/anthology/D16-1244`.

T. Parsons. *Events in the Semantics of English: A study in subatomic semantics*. MIT Press, Cambridge, MA, 1990.

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. 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 8024–8035. Curran Associates, Inc., 2019. URL `http://papers.neurips.cc/paper/`

9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

M. Pazzani, C. Brunk, and G. Silverstein. A knowledge-intensive approach to learning relational concepts. In *Proceedings of the Eighth International Workshop on Machine Learning*, pages 432–436, Evanston, IL, 1991.

B. Peng, Z. Lu, H. Li, and K. Wong. Towards neural network-based reasoning. *CoRR*, abs/1508.05508, 2015.

J. Pennington, R. Socher, and C. D. Manning. GloVe: Global vectors for word representation. In *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. URL http://www.aclweb.org/anthology/D14-1162.

B. Percha, Y. Garten, and R. B. Altman. Discovery and explanation of drug-drug interactions via text mining. In *Pacific symposium on biocomputing*, page 410. NIH Public Access, 2012.

C. Perfetti, J. Van Dyke, and L. Hart. The psycholinguistics of basic literacy. *Annual Review of Applied Linguistics*, 21:127 – 149, 01 2001. doi: 10.1017/S0267190501000083.

M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018.

F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, and A. Miller. 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, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1250. URL https://www.aclweb.org/anthology/D19-1250.

J. Piaget. *The Language and Thought of the Child, trans.* Harcourt, Brace, 1926.

J. Pino and M. Eskénazi. Semi-automatic generation of cloze question distractors effect of students' l1. In *SLaTE*, pages 65–68. ISCA, 2009.

J. Pino, M. Heilman, and M. Eskenazi. A Selection Strategy to Improve Cloze Question Quality. In *Proceedings of the Workshop on Intelligent Tutoring Systems for Ill-Defined Domains. 9th International Conference on Intelligent Tutoring Systems.*, 2008.

A. Poliak, J. Naradowsky, A. Haldar, R. Rudinger, and B. Van Durme. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191. Association for Computational Linguistics, 2018. doi: 10.18653/v1/S18-2023. URL `http://aclweb.org/anthology/S18-2023`.

C. Qin, K. Dvijotham, B. O'Donoghue, R. Bunel, R. Stanforth, S. Gowal, J. Uesato, G. Swirszcz, and P. Kohli. Verification of non-linear specifications for neural networks. *CoRR*, abs/1902.09592, 2019.

J. R. Quinlan. Learning logical definitions from relations. *Machine Learning*, 5: 239–266, 1990.

A. Raghunathan, J. Steinhardt, and P. Liang. Certified defenses against adversarial examples. *arXiv preprint arXiv:1801.09344*, 2018a.

A. Raghunathan, J. Steinhardt, and P. Liang. Semidefinite relaxations for certifying robustness to adversarial examples. In *Advances in Neural Information Processing Systems*, pages 10877–10887, 2018b.

P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. 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, Nov. 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL `https://www.aclweb.org/anthology/D16-1264`.

P. Rajpurkar, R. Jia, and P. Liang. 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, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2124. URL `https://www.aclweb.org/anthology/P18-2124`.

M. T. Ribeiro, S. Singh, and C. Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Knowledge Discovery and Data Mining (KDD)*, 2016.

M. T. Ribeiro, S. Singh, and C. Guestrin. Anchors: High-precision model-agnostic explanations. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018a.

M. T. Ribeiro, S. Singh, and C. Guestrin. Semantically equivalent adversarial rules for debugging NLP models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865, Melbourne, Australia, July 2018b. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/P18-1079`.

B. L. Richards and R. J. Mooney. First-order theory revision. In *Proceedings of the Eighth International Workshop on Machine Learning*, pages 447–451, Evanston, IL, 1991.

M. Richardson and P. Domingos. Markov logic networks. *Mach. Learn.*, 62(1-2):107–136, 2006. ISSN 0885-6125. doi: 10.1007/s10994-006-5833-1. URL `http://dx.doi.org/10.1007/s10994-006-5833-1`.

M. Richardson, C. J. Burges, and E. Renshaw. MCTest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 193–203, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/D13-1020`.

S. Riedel, L. Yao, and A. McCallum. Modeling relations and their mentions without labeled text. In *Proceedings of the 2010 European Conference on Machine*

Learning and Knowledge Discovery in Databases: Part III, ECML PKDD'10, pages 148–163, 2010. ISBN 3-642-15938-9, 978-3-642-15938-1.

S. Riedel, L. Yao, A. McCallum, and B. M. Marlin. Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 74–84, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/N13-1008`.

J. J. Rocchio. Relevance feedback in information retrieval. In G. Salton, editor, *The SMART Retrieval System - Experiments in Automatic Document Processing*. Prentice Hall, 1971.

T. Rocktäschel and S. Riedel. End-to-end differentiable proving. *Advances in Neural Information Processing Systems 30*, pages 3788–3800, 2017. URL `http://papers.nips.cc/paper/6969-end-to-end-differentiable-proving.pdf`.

T. Rocktäschel, E. Grefenstette, K. M. Hermann, T. Kocisky, and P. Blunsom. Reasoning about entailment with neural attention. In *International Conference on Learning Representations (ICLR)*, 2016.

D. Rumelhart and A. Ortony. *The Representation of Knowledge in Memory*, pages 99–135. Hillsdale, NJ: Erlbaum, 1977.

M. Sachan, A. Dubey, and E. P. Xing. Science question answering using instructional materials. *CoRR*, abs/1602.04375, 2016. URL `http://arxiv.org/abs/1602.04375`.

A. Saha, R. Aralikatte, M. M. Khapra, and K. Sankaranarayanan. DuoRC: Towards Complex Language Understanding with Paraphrased Reading Comprehension. In *Meeting of the Association for Computational Linguistics (ACL)*, 2018.

K. Sakaguchi, Y. Arase, and M. Komachi. Discriminative approach to fill-in-the-blank quiz generation for language learners. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 2: Short Papers*, pages 238–242, 2013. URL http://aclweb.org/anthology/P/P13/P13-2043.pdf.

G. Salton and C. Buckley. Readings in information retrieval. In K. Sparck Jones and P. Willett, editors, *""*, chapter Improving Retrieval Performance by Relevance Feedback, pages 355–364. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997. ISBN 1-55860-454-5. URL http://dl.acm.org/citation.cfm?id=275537.275712.

R. C. Schank and R. P. Abelson. Scripts, plans, and knowledge. In *Proceedings of the 4th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'75, page 151–157, San Francisco, CA, USA, 1975. Morgan Kaufmann Publishers Inc.

L. Schmidt, S. Santurkar, D. Tsipras, K. Talwar, and A. Madry. Adversarially robust generalization requires more data. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 5014–5026. Curran Associates, Inc., 2018. URL http://papers.nips.cc/paper/7749-adversarially-robust-generalization-requires-more-data.pdf.

C. Schoenick, P. Clark, O. Tafjord, P. Turney, and O. Etzioni. Moving beyond the turing test with the allen ai science challenge. *arXiv preprint arXiv:1604.04315*, 2016.

S. Schoenmackers, O. Etzioni, and D. S. Weld. Scaling textual inference to the web. In *EMNLP '08: Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 79–88, 2008. URL http://portal.acm.org/citation.cfm?id=1613727.

S. Schoenmackers, O. Etzioni, D. S. Weld, and J. Davis. Learning first-order horn clauses from web text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 1088–1098, 2010. URL `http://dl.acm.org/citation.cfm?id=1870658.1870764`.

R. Schwartz, M. Sap, I. Konstas, L. Zilles, Y. Choi, and N. A. Smith. The effect of different writing tasks on linguistic style: A case study of the ROC story cloze task. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 15–25, 2017.

I. Segura-Bedmar, P. Martínez, and M. Herrero Zazo. SemEval-2013 Task 9: Extraction of drug-drug interactions from biomedical texts (DDIExtraction 2013). In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 341–350, 2013.

M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi. Bidirectional attention flow for machine comprehension. In *The International Conference on Learning Representations (ICLR)*, 2017a.

M. Seo, S. Min, A. Farhadi, and H. Hajishirzi. Query-reduction networks for question answering. *ICLR*, 2017b.

Y. Shen, P.-S. Huang, J. Gao, and W. Chen. ReasoNet: Learning to stop reading in machine comprehension. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, pages 1047–1055, 2017. ISBN 978-1-4503-4887-4. doi: 10.1145/3097983.3098177. URL `http://doi.acm.org/10.1145/3097983.3098177`.

R. Snow, B. O'Connor, D. Jurafsky, and A. Ng. Cheap and fast – but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 254–263, Honolulu, Hawaii, Oct. 2008. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/D08-1027`.

A. Sordoni, P. Bachman, and Y. Bengio. Iterative alternating neural attention for machine reading. *CoRR*, abs/1606.02245, 2016.

M. Steedman. *Surface structure and interpretation*, volume 30 of *Linguistic inquiry*. MIT Press, 1996.

P. Stenetorp, G. Topić, S. Pyysalo, T. Ohta, J.-D. Kim, and J. Tsujii. BioNLP shared task 2011: Supporting resources. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 112–120, 2011.

S. Sugawara, K. Inui, S. Sekine, and A. Aizawa. What makes reading comprehension questions easier? In *EMNLP*, pages 4208–4219. Association for Computational Linguistics, 2018.

S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus. End-to-end memory networks. In *Advances in Neural Information Processing Systems*, pages 2440–2448, 2015.

E. Sumita, F. Sugaya, and S. Yamamoto. Measuring non-native speakers' proficiency of english by using a test with automatically-generated fill-in-the-blank questions. In *Proceedings of the Second Workshop on Building Educational Applications Using NLP*, EdAppsNLP 05, pages 61–68, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=1609829.1609839`.

C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

A. Talmor and J. Berant. 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, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1059. URL `https://www.aclweb.org/anthology/N18-1059`.

A. Talmor and J. Berant. 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, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1485. URL `https://www.aclweb.org/anthology/P19-1485`.

A. Talmor, Y. Elazar, Y. Goldberg, and J. Berant. olmpics – on what language model pre-training captures, 2019.

Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May 2016. URL `http://arxiv.org/abs/1605.02688`.

J. Thorne and A. Vlachos. Adversarial attacks against fact extraction and verification. *CoRR*, abs/1903.05543, 2019. URL `http://arxiv.org/abs/1903.05543`.

J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal. FEVER: a large-scale dataset for fact extraction and verification. In *NAACL-HLT*, 2018.

A. Trischler, T. Wang, X. Yuan, J. Harris, A. Sordoni, P. Bachman, and K. Suleman. NewsQA: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200, Vancouver, Canada, Aug. 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-2623. URL `https://www.aclweb.org/anthology/W17-2623`.

D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=SyxAb30cY7`.

J. Uesato, B. O'Donoghue, A. v. d. Oord, and P. Kohli. Adversarial risk and the dangers of evaluating against weak attacks. *arXiv preprint arXiv:1802.05666*, 2018.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017. URL `http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf`.

D. Vrandečić. Wikidata: A new platform for collaborative data collection. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12 Companion, pages 1063–1064, 2012. ISBN 978-1-4503-1230-1. doi: 10.1145/2187980.2188242. URL `http://doi.acm.org/10.1145/2187980.2188242`.

E. Wallace, S. Feng, N. Kandpal, M. Gardner, and S. Singh. Universal adversarial triggers for NLP. *CoRR*, abs/1908.07125, 2019a.

E. Wallace, P. Rodriguez, S. Feng, I. Yamada, and J. Boyd-Graber. Trick me if you can: Human-in-the-loop generation of adversarial question answering examples. *Transactions of the Association of Computational Linguistics*, 10, 2019b.

C. Wang, R. Bunel, K. Dvijotham, P.-S. Huang, E. Grefenstette, and P. Kohli. Knowing when to stop: Evaluation and verification of conformity to output-size specifications. *arXiv preprint arXiv:1904.12004*, 2019.

S. Wang, M. Yu, J. Jiang, W. Zhang, X. Guo, S. Chang, Z. Wang, T. Klinger, G. Tesauro, and M. Campbell. Evidence aggregation for answer re-ranking in open-domain question answering. *CoRR*, abs/1711.05116, 2017.

S. Wang, K. Pei, J. Whitehouse, J. Yang, and S. Jana. Formal security analysis of neural networks using symbolic intervals. *arXiv preprint arXiv:1804.10829*, 2018a.

S. Wang, M. Yu, X. Guo, Z. Wang, T. Klinger, W. Zhang, S. Chang, G. Tesauro, B. Zhou, and J. Jiang. R^3: Reinforced ranker-reader for open-domain question answering. In S. A. McIlraith et al., editors, *Proceedings of the Thirty-Second*

AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), pages 5981–5988. AAAI Press, 2018b.

Y. Wang and M. Bansal. Robust machine comprehension models via adversarial training. 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 575–581, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2091. URL `https://www.aclweb.org/anthology/N18-2091`.

D. Weissenborn, G. Wiese, and L. Seiffe. 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, Aug. 2017. Association for Computational Linguistics. doi: 10.18653/v1/K17-1028. URL `https://www.aclweb.org/anthology/K17-1028`.

J. Welbl, N. F. Liu, and M. Gardner. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 94–106, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4413. URL `https://www.aclweb.org/anthology/W17-4413`.

J. Welbl, P. Stenetorp, and S. Riedel. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302, 2018. doi: 10.1162/tacl_a_00021. URL `https://www.aclweb.org/anthology/Q18-1021`.

J. Welbl, P.-S. Huang, R. Stanforth, S. Gowal, K. D. Dvijotham, M. Szummer, and P. Kohli. Towards verified robustness under text deletion interventions. In *International Conference on Learning Representations*, 2020a. URL `https://openreview.net/forum?id=SyxhVkrYvr`.

J. Welbl, P. Minervini, M. Bartolo, P. Stenetorp, and S. Riedel. Undersensitivity in neural reading comprehension, 2020b.

T.-W. Weng, H. Zhang, H. Chen, Z. Song, C.-J. Hsieh, D. Boning, I. S. Dhillon, and L. Daniel. Towards fast computation of certified robustness for relu networks. *arXiv preprint arXiv:1804.09699*, 2018.

J. Weston, S. Chopra, and A. Bordes. Memory networks. *ICLR*, 2015.

J. Weston, A. Bordes, S. Chopra, and T. Mikolov. Towards AI-complete question answering: A set of prerequisite toy tasks. *ICLR*, 2016.

A. Williams, N. Nangia, and S. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122. Association for Computational Linguistics, 2018. URL `http://aclweb.org/anthology/N18-1101`.

E. Wong and Z. Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, pages 5283–5292, 2018.

C. Xiong, V. Zhong, and R. Socher. Dynamic coattention networks for question answering. *ICLR*, 2017.

J. Xu and W. B. Croft. Query expansion using local and global document analysis. In H. Frei et al., editors, *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'96*, pages 4–11. ACM, 1996.

S. Yagcioglu, A. Erdem, E. Erdem, and N. Ikizler-Cinbis. RecipeQA: A challenge dataset for multimodal comprehension of cooking recipes. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1358–1368, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational

Linguistics. doi: 10.18653/v1/D18-1166. URL https://www.aclweb.org/anthology/D18-1166.

Y. Yang, W.-t. Yih, and C. Meek. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018, Lisbon, Portugal, Sept. 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1237. URL https://www.aclweb.org/anthology/D15-1237.

Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. Cohen, R. Salakhutdinov, and C. D. Manning. 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. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/D18-1259.

A. W. Yu, D. Dohan, Q. Le, T. Luong, R. Zhao, and K. Chen. Fast and accurate reading comprehension by combining self-attention and convolution. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=B14TlG-RW.

J. M. Zelle and R. J. Mooney. Learning to parse database queries using inductive logic programming. In W. J. Clancey and D. S. Weld, editors, *AAAI/IAAI, Vol. 2*, pages 1050–1055. AAAI Press / The MIT Press, 1996.

R. Zellers, Y. Bisk, R. Schwartz, and Y. Choi. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1009. URL https://www.aclweb.org/anthology/D18-1009.

T. Zesch and O. Melamud. Automatic generation of challenging distractors using context-sensitive inference rules. In *Proceedings of the Ninth Workshop on Innovative Use of NLP for Building Educational Applications, BEA@ACL*

2014, June 26, 2014, Baltimore, Maryland, USA, pages 143–148, 2014. URL http://aclweb.org/anthology/W/W14/W14-1817.pdf.

L. S. Zettlemoyer and M. Collins. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, UAI'05, page 658–666, Arlington, Virginia, USA, 2005. AUAI Press. ISBN 0974903914.

W. E. Zhang, Q. Z. Sheng, and A. A. F. Alhazmi. Generating textual adversarial examples for deep learning models: A survey. *CoRR*, abs/1901.06796, 2019.

Z. Zhao, D. Dua, and S. Singh. Generating natural adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2018.

H. Zhu, L. Dong, F. Wei, W. Wang, B. Qin, and T. Liu. Learning to ask unanswerable questions for machine reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4238–4248, Florence, Italy, July 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P19-1415.

G. K. Zipf. *The Psychobiology of Language*. Houghton-Mifflin, New York, NY, USA, 1935.

R. A. Zwaan, M. C. Langston, and A. C. Graesser. The construction of situation models in narrative comprehension: An event-indexing model. *Psychological Science*, 6(5):292–297, 1995. doi: 10.1111/j.1467-9280.1995.tb00513.x. URL https://doi.org/10.1111/j.1467-9280.1995.tb00513.x.