Text Classification

Gaurav Singh

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

Department of Computer Science
University College London

September 16, 2019

I, Gaurav Singh, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

There is an abundance of text data in this world but most of it is raw. We need to extract information from this data to make use of it. One way to extract this information from raw text is to apply informative labels drawn from a pre-defined fixed set i.e. Text Classification. In this thesis, we focus on the general problem of text classification, and work towards solving challenges associated to binary/multi-class/multi-label classification. More specifically, we deal with the problem of (i) Zero-shot labels during testing; (ii) Active learning for text screening; (iii) Multi-label classification under low supervision; (iv) Structured label space; (v) Classifying pairs of words in raw text i.e. Relation Extraction.

For (i), we use a zero-shot classification model that utilizes independently learned semantic embeddings. Regarding (ii), we propose a novel active learning algorithm that reduces problem of bias in naive active learning algorithms. For (iii), we propose neural candidate-selector architecture that starts from a set of high-recall candidate labels to obtain high-precision predictions. In the case of (iv), we proposed an attention based neural tree decoder that recursively decodes an abstract into the ontology tree. For (v), we propose using second-order relations that are derived by explicitly connecting pairs of words via context token(s) for improved relation extraction.

We use a wide variety of both traditional and deep machine learning tools. More specifically, we used traditional machine learning models like multi-valued linear regression and logistic regression for (i, ii), deep convolutional neural networks for (iii), recurrent neural networks for (iv) and transformer networks for (v).

Impact Statement

The work in this thesis has been focused on the problem extracting information from unstructured (or raw) text. This information can be extracted in multiple ways but all of those ways involve labeling the unstructured data with pre-defined informative labels. In this thesis we have tackled some of the issues encountered in automating this process. Despite the fact that the proposed methods in this work are general enough to be used in any field, we apply ourselves to the task of extracting information from biomedical text e.g. biomedical abstracts or doctor's notes. These methods can have significant implications for academic research, public health and commercial entities.

There has been a recent push in the academic community for the use of AI in healthcare, one possible use-case for which is extracting information from millions of published biomedical papers. The methods proposed in this thesis can be used to automate this process of extracting information that can be used in wide-ranging applications, including but not limited to drug discovery and systematic review. Our novel use of software (e.g. Metamap) for more accurate structured text tagging can also help the academic research community to handle low-resource tasks that at times plague biomedical research problems.

The research presented in this work has the potential to make an impact on the general public health as well. An important prerequisite to formulating public health guidelines is to conduct Systematic Reviews. The process of conducting these reviews at present requires extensive human-effort, making it an expensive process, which in turn limits its use. We have made significant progress in making this process more automated. We think this can lead to reduced cost for systematic reviews in the

future, making the assessment of different health interventions feasible, in addition to other things.

The work in this thesis not only impacts the academic research community and public health, as mentioned above, but can also make an impact commercially. The superior text-tagging can be used to assist in the process of drug-discovery, to extract important bits of structured information from millions of freely available biomedical papers, that can then be used to come up with better candidate drugs. In fact, one of our work on relation extraction is of particular interest to multiple health-tech companies working towards drug-discovery.

Acknowledgements

I would like to acknowledge the financial support received from Faculty of Engineering at UCL through Dean's award, FRE program at Yahoo! (via Prof. John Shawe-Taylor) and Transform project at Cochrane (via Prof. James Thomas). I would like to thank and acknowledge with gratitude the support that I received from both supervisors, Prof. John Shawe-Taylor (HoD and Professor at CS Dept, UCL) and Prof. James Thomas (Professor at Institute of Education, UCL). Also, I would like to thank each one of my co-authors, especially Prof. Byron C. Wallace (Asst. Professor CS Dept, Northeastern University), whose constant support and advice was instrumental in the completion of this thesis, and from whom I learnt a lot regarding research during these years.

Publications

1. Neighborhood Sensitive Mapping for Zero-Shot Classification using Independently Learned Semantic Embeddings, ACML Workshop on Learning on Big Data, 2016
   Authors: Gaurav Singh, Fabrizio Silvestri and John Shawe-Taylor

2. Improving Active Learning in Systematic Reviews, arXiv:1801.09496
   Authors: Gaurav Singh, James Thomas and John Shawe-Taylor

3. A Neural Candidate-Selector Architecture for Automatic Structured Clinical Text Annotation, CIKM 2017
   Authors: Gaurav Singh, Iain J. Marshall, James Thomas, John Shawe-Taylor and Byron C. Wallace

4. Identifying Diagnostic Test Accuracy Publications using a Deep Model, CLEF (*Working Notes*) 2017
   Authors: Gaurav Singh, James Thomas, Iain J. Marshall and Byron C. Wallace

5. Structured Multi-Label Biomedical Text Tagging via Attentive Neural Tree Decoding, EMNLP 2018
   Authors: Gaurav Singh, James Thomas, Iain J. Marshall, John Shawe-Taylor and Byron C. Wallace

6. Relation Extraction using Explicit Context Conditioning, NAACL 2019
   Authors: Gaurav Singh and Parminder Bhatia

Contents

List of Figures

List of Tables

Chapter 1

Introduction

There is a lot of text available over the internet. Albeit most of it is in the form of unstructured raw data. This unstructured text contains a lot of useful information that needs to be manually extracted, but this manual process can be expensive. Hence there is a need for methods that do automatic information extraction. One such method involves applying labels that encode the details contained in the text. These labels can then be used for tasks like *categorizing the text*, *summarizing the content* and *easing searchability*. This task of assigning labels to free text is called *Text Classification*.

The labels are drawn from a pre-defined set. This pre-defined set is usually hand-crafted based on the task at hand and/or the knowledge of the domain. The size of the label set depends on the coarseness of the task. If the information that needs to be extracted is general then the label set is usually small. In fact when the label set contains just two labels it is referred to as *Binary Classification*. When it contains more than two labels it is called *Multi-class Classification*. One more category is *Multi-label Classification* or *text-tagging* that refers to the situation where multiple labels are applied to a single piece of text. In this thesis we deal with different aspects of text classification, few of which are:

- If the training data does not contain all the labels defined in the label set, can we still predict those zero-shot labels?
- If the training data is small and the output space vast and structured, can we still perform multi-label classification?

- Can we leverage the ontology better than other methods for hierarchical classification?

- Can we do better than directly using context representations generated by encoders for relation extraction?

1.1 Applications

There is no dearth of unstructured text in the age of the internet. There are thousands of papers published every year that need to be summarized, informative blog-posts that need to be remembered, millions of posts shared every day over Facebook/Twitter/LinkedIn that contain useful information, websites that need to be tagged/bookmarked. All of these tasks can be done using text classification. We can not overstate the usefulness of an accurate text classification system. Therefore, we have focused our attention on some of the specific aspects of text classification. We mention some of the many applications of text classification below:

- **Zero-Shot Text Classification**. At times the training data needs to be manually labeled. This makes it expensive to obtain large quantities of labeled training data. This leads to the problem of unseen labels, especially when the label space is large, since not all of the labels can be found in a small train set. This is a problem in a number of areas that require expensive manual annotations like biomedical/clinical text tagging, URL bookmarking, paper abstract categorization etc.

- **Structured Text Tagging**. There are times when the output space is structured which means that the labels are related to each other via some well defined structure. For example, in clinical text annotations the labels might come from a medical ontology, or the labels might be organized in different categories like PICO (standing for Population, Intervention/Control and Outcomes).

- **Citation Screening**. Systematic review is the task of summarizing findings of different clinical and social science studies. While systematic reviews are mostly applied in the biomedical or healthcare context, they can be used in

other areas where an assessment of a precisely defined subject would be helpful. Having said that, to the best of our knowledge, they are not used in Computer Science. The first step of a systematic review task is to identify all the studies that are relevant to the review in question, and need to be summarized. This is basically the binary classification of abstracts into relevant and non-relevant categories. It is an essential first step in summarizing the studies for various purposes e.g. *policy formulation*. In fact, this can be compared to the problem of searching for documents relevant to a given query - a well-studied problem in the field of Information Retrieval -, the only difference being that the query in this case is the review topic, and it is fixed for a given review.

- **Text Summarization**. Text tagging can be used for summarizing the contents of an unstructured piece of text e.g. yelp reviews, clinical text, etc. The unstructured piece of text can be tagged with very specific labels that are hand-crafted to summarize the information contained therein.

1.2 Motivations

Text tagging requires labeled data to train the classification model. In fact, it would not be wrong to say that the learned model gets better with the quantity of data available for training. But obtaining labeled data is expensive and time consuming, which means that when labeled data is small then there would be plenty of labels that would not be observed during training. This led to our first work on zero-shot classification (ZSC) for text using independently learned semantic embeddings. This can be especially useful in clinical text tagging due to expensive nature of labeled data and missing labels. While most works on zero-shot classification at the time were focused on images [5, 6], we explore it for text. Though, there have been some works on zero-shot text classification in the recent past [7, 8].

Text classification has proved useful in the field of systematic reviews (SRs). These SRs are used to summarize the results of different clinical and social science studies. The first step in a systematic review is to identify all the studies relevant to the review question, unfortunately, this task is still performed manually. Lately,

there have been some attempts to reduce this manual effort using active learning. In active learning the model is updated after every few iterations using newly labeled data. It works by selecting a small number of documents that are manually annotated after each iteration and these are then used to train the classifier. Documents to be annotated are selected based on the relevance scores generated by the classifier. We noticed that this way of selecting documents would bias the classifier early on into selecting documents that are similar to the ones it was trained on during initial iterations. One straightforward way of reducing this bias is to perform uncertainty based sampling, but unfortunately, only naive active learning algorithm has been used for citation screening in SRs. Therefore, we introduce a novel active learning algorithm that would help the classifier explore different topics, as opposed to simply selecting the most relevant documents or randomly sampling studies that the classifier is uncertain on. Also, we find that simple feature extraction models perform well for the task of systematic reviews contrary to the claims of a previously published results [9].

PICO (Population, Intervention/Control and Outcome) annotations are an important application of structured multi-label text tagging. It involves the task of automatically annotating free clinical texts with concepts that describe complimentary (yet, correlated) clinically salient aspects of the underlying trials. This specific problem poses a few unique challenges. One issue was the size ($\approx$300K labels) of the output space due to a large medical vocabulary. On top of that, the lack of annotated data in this domain makes the task even more harder. We propose a neural candidate-selector architecture that heuristically generates a high-recall list of candidate concepts and then filters it out to obtain the final high-precision predictions. This approaches leverages external information in the form of a software called Metamap [10].

At times labels are drawn from pre-defined tree structure (i.e., ontology) for multi-label text tagging. There are approaches that can indirectly leverage these structures in the label space using either node embeddings trained on the structure or inducing a graph structure in the output space during learning. But none of the

previous approaches directly used the given ontology. Hence we design a seq-to-tree model that takes as input a piece of free text and decodes into a tree-structured ontology. It leads to predictions that are more coherent with the ontology.

The task of classifying pairs of words into their relation type is called Relation Extraction (RE). Apart from tagging text with informative labels, relations between different words contained therein can give structured information. We propose a model that utilizes indirect relations between words, in addition to, more direct relations that are computed by traditional state-of-the-art models. We get an indirect relation between two words by connecting them via a context token, and call it a second-order relation. For comparison, some past works have used a pre-defined list of trigger words for extracting relations [11, 12], but these trigger words are relation/task/data/language specific. They also have to be hand designed and are often not exhaustive. In contrast, we do not require a pre-defined list of trigger words, but train the model to infer the connecting word given an unstructured piece of text.

1.3 Contributions

Our focus in this work is on the general task of text classification. We deal with different aspects of text classification like: 1) Zero-shot labels; 2) Insufficient labeled data for training; 3) Text tagging using structured ontology; 4) Relation Extraction using Explicit Context Conditioning. Here we enlist the specific contributions that we made during our efforts to solve the above problems.

1.3.1 Zero-shot Text Classification

A traditional classifier can not assign zero-shot labels due to absence of information about them. Zero-shot classifier remedies that by using independently learned semantic representations for those labels. These embeddings help establish a relation between the seen and unseen labels, which can then be used to assign a zero-shot label to an instance based on its similarity to the seen labels, and the similarities between the seen and unseen labels. Hence the quality of these semantic embeddings is crucial to the task of zero-shot classification. Unfortunately, as these embeddings have been trained on an unrelated task, therefore, they can be extremely noisy. To

address this, we re-train these embeddings using the labeled training data for the task at hand, before using them for zero-shot classification. It leads to an improvement in the results when compared with using independently learned embeddings.

1.3.2 Text Classification for Citation Screening

The task of summarizing the contents of clinical and social science studies is known as Systematic Reviews (SR). The first step in a SR is to extract studies relevant to the topic at hand, and this is done using active learning [13, 14, 15, 16]. Traditionally, these active learning methods for citation screening start from a small set of annotated documents to train a classifier. Afterwards, they iteratively select k documents that get the highest scores for relevance from the classifier. These documents are then manually annotated and used to retrain the classifier. But this method of document selection would get biased towards giving higher scores to documents that are similar to the previously annotated ones. We proposed a method that explores different topics during the beginning of the active learning process. It can be said that our method sacrifices performance in the beginning to explore the topic space, which leads to better results in terms of work-saved by the end of the process. While systematic reviewing can used for any field of work that requires an assessment of a precise subject, to the best of our knowledge, they are not used in Computer Science.

1.3.3 Automatic Structured Clinical Text Annotation

We consider the task of automatic structured clinical text annotation. This important problems comes with specific challenges of its own. Those challenges are: the lack of sufficient labeled data, vast output space, correlated labels contained in disjoint sets and zero-shot labels. We propose a model that achieves good results over a clinical dataset under the constraint of these problems. The specific contribution of this work was a novel method for multilabel classification into multiple distinct, but correlated label sets using a neural model that considers ‘candidate’ label tuples, conditioned on the text being annotated. Our approach addresses training data sparsity by reframing the annotation task as a two step process in which we first generate a set of candidate annotations relevant to the input text, and then we select and group these annotations.

In our case, we generate candidates using both (a) a multitask model directly trained to generate candidate concepts, and, (b) the *MetaMap* tool [10]. We then use a neural discriminative model to infer plausible triplets of concepts from the unstructured candidate set, conditioned on the free-text being annotated.

1.3.4 Structured Text Tagging from Ontology

We focus on the task of assigning labels drawn from a tree structured vocabulary, i.e an ontology. These tree structured vocabs can be an important source of information and need to be leveraged. Attempts have been made in the past to use this structure for superior text tagging, but these attempts have been focused on learning structure aware representations for tags or induce a tree structure in the output space. We try to directly use the given ontology by training a decoder to traverse the tree structure downwards from the root activating and recursively expanding pertinent child nodes/tags. We should mention that the proposed method only works for ontologies that are strictly tree-structured and not DAGs.

1.3.5 Relation Extraction using Explicit Context Conditioning

We focus on the task of labeling relations between groups of entities in text, known as Relation Extraction. Most current RE models learn context-aware representations of the target entities that are then used to establish relation between them. This works well for intra-sentence RE, and we call them first-order relations. However, this methodology can sometimes fail to capture complex and long dependencies. To address this, we hypothesize that at times two target entities can be explicitly connected via a context token. We refer to such indirect relations as second-order relations, and describe an efficient implementation for computing them. These second-order relation scores are then combined with first-order relation scores. We should mention here that our model identifies the connecting context tokens based on the input text, in contrast to using a pre-defined list of trigger words [11, 12] that indicate existence of a relation between two entities. Our empirical results show that the proposed method leads to state-of-the-art performance over two biomedical datasets.

1.4 Literature Review

There has been an abundance of data ever since the Internet came into existence, lots of data (i.e. image/text/audio/video) is generated every second, and instantly shared across the world. The need for automatic information extraction from this unstructured raw data is what provided motivation for advancements in the field of machine learning (ML) and pattern recognition [17]. There are two broad categories of problems that have received a lot of attention in machine learning: 1) Classification [18] and 2) Regression [19]. In this thesis we focus on the problem of classification, more specifically text classification, which requires assigning label(s) to input from a pre-defined set of labels. This is in contrast to regression that requires predicting a real number as target e.g. the amount of rain on a day or the age of a person.

The first few ML classification models were based on recognizing patterns in the data using clustering [20]. They would basically cluster similar data points and assign any new sample to the cluster that most resembled it. It was assumed that the sample was an average (mean) of data points in that cluster in many aspects. There were also some works on using nearest neighbours (NN) for classification [21], where the new sample was considered an average of its nearest neighbours. These nearest neighbours were computed based on some distance metric defined over the space of input samples e.g. euclidean space.

Afterwards, tree based ML models e.g. Gradient Boosted Decision Trees [22] or Random Forest Classifier [23], achieved lots of success and gained popularity. At the same time, there was a shift towards more sophisticated non-linear models to divide data points into distinct groups based on there similarities in a projected space e.g. Kernel Hilbert Space. These machines used support vectors to draw a margin between distinct groups, and were called Support Vector Machines [24, 25]. These models provided excellent theoretical guarantees on convergence [26] and generalization [27], and were widely used in the past. There was also some research on deep learning [28, 29, 30] but they did not receive a lot of attention at the time. More recently, there has been a shift towards deep neural networks for machine learning [31, 32]. These deep neural networks have achieved state-of-the-art performance in

many natural language understanding tasks [33, 34, 35].

In this thesis, we have focused on different types of text classification e.g. multi-class or multi-label [36, 37], and problems associated with these types e.g. lack of labeled data [38] or structured output space [39], using both traditional machine learning and newer deep learning based models. We dedicate a separate chapter for every problem (or group of problems) associated with these issues. Despite the fact that most of these chapters can be categorized under text classification, they still required (slightly) differing literature reviews. **For this reason, we provide a more detailed literature review in each chapter for the specific problem(s) discussed in that chapter.**

1.5 Organization

The rest of the thesis is organized as follows.

Chapter 2. We discusses our work on zero-shot text classification.

Chapter 3. We discusses our work on active learning for binary text classification for the task of citation screening.

Chapter 4. We discuss our work on automatic structured clinical text annotation i.e. PICO (Population, Intervention/Control, Outcome) annotations.

Chapter 5. We discuss our work on biomedical text tagging when the labels are drawn from a tree structured vocabulary i.e. *Ontology*.

Chapter 6. We present our work on relation extraction (RE) from unstructured text. While text classification involves classifying sequences of words, RE is the task of classifying groups of words into the relations between them.

Chapter 7. Then we present the conclusions of our progress till now, the gaps still left in the literature and discuss our proposals for fixing them.

Chapter 2

Neighborhood Sensitive Mapping for Zero-Shot Classification

One of the most prominent areas of research in machine learning is *classification*. In a traditional setting, the problem of classification consists of training a model to approximate a target function $f : \mathscr{X} \rightarrow \mathscr{Y}$ where at least a sample for each $y \in \mathscr{Y}$ is presented to the training algorithm. A common problem faced by many systems, especially in their early stage is the absence of training instances for all possible classes. In such cases, a traditional classifier cannot, in fact, assign class labels that have not been observed in the training set. This is one of the main reasons for the growth of a research area, *zero-shot classification*, studying how classification can be done also using unseen labels. In a zero-shot setting [6], we are given a subset $\hat{\mathscr{Y}} \subset \mathscr{Y}$ of the labels for which we do not observe any corresponding training instance. Still, the function f that we train must be able to correctly assign labels also on $\hat{\mathscr{Y}}$. There is a growing need for using classification systems in order to automate different online and offline tasks, often having labels that are yet to be observed in the training data. Based on the definitions given by Palatucci *et al.* [6], we address the following general research question: "*Given a semantic encoding of a large set of concept labels, can we build a classifier to recognize labels that were omitted from the training set?*"

To correctly deal with unseen labels, one possibility is to establish relationship between seen and unseen labels. One of the earliest works in zero-shot classification

[6] proposes a method based on creating semantic embeddings for words based on co-occurrences of labels in dictionaries and human feedback on label properties. More recently, the progress made in learning semantic embeddings (e.g. word2vec [40]) has provided a method for encoding the semantic meaning of words. Therefore, classification tasks where the label set is made up of meaningful words can be used to establish such inter-label relationships. An input is mapped to the label embedding space, and then, a nearest neighbors approach is used to predict the correct label, including from those not seen in the training set. Unfortunately, these methods that are aimed at "directly learning" a mapping from the input to a semantic space may suffer from two major problems.

First, the target semantic space may be hard to learn due to the noise. This is because similarity between word embeddings of two labels may not correspond to the similarity between their respective inputs. Such a situation can occur because the word embeddings for labels were constructed using an independent dataset (i.e. google news). As an example, if two words (labels) occur together in google news corpora - that was used to learn the word embeddings - then they would have similar embedding representation, with no regards to their similarity in the input space. Hence, semantic space representations (i.e. word embeddings) for labels can vary significantly based on the dataset used to learn them, and this could be a problem in zero-shot classification.

Second, nearest neighbor classifiers for zero-shot classification are not designed to take into account the neighborhood of a given label in the embedding space. This basically means that labels that have really close neighbors should be learned with more accuracy at the expense of labels that are isolated in their neighborhood.

We address the two above mentioned problems in a step-wise manner. Firstly, we need to fine-tune label embeddings based on the task. Secondly, we need to develop a neighborhood sensitive mapping that can reduce the risk of error for a nearest neighbor classifier during zero-shot classification.

Our main contributions in this study are:

1. Learning data-dependent embeddings for labels

2. Neighborhood sensitive mapping to reduce classification error

2.1 Related Works

There has been active research in the area of zero-shot classification in the recent past. Originally, the problem was defined in its current form by Palatucci *et al.* [6], where they address the problem by embedding the set of labels into a semantic space used then to extrapolate information about unseen labels. Given an object in the dataset, they firstly predict the set of semantic features (in the embedding space) corresponding to that input, and then they find the nearest class in the labels embedding. They also develop a generalization bound on the error for zero-shot classification using a nearest neighbour classifier. A following work [41] proposed a multi-label max-margin classifier with applications to zero-shot classification. By means of a correlation matrix between different labels they are able to predict unseen labels. The method, specifically designed for multi-label classification, explicitly reduces the hamming loss between prediction vectors and the label vectors.

There are numerous practical applications where unseen images need to be classified. As a result, zero-shot classification has found considerable interest in the area of image recognition and classification. Another such work [42] develops an approach specifically based on learning attribute for animals (e.g. color, eating habits etc.). These attributes are not unique to a single animal, and therefore, can be learned from the available training data. One more work [43] proposed a method specifically targeting images, that focuses on exploiting co-occurrences of visual concepts in images for knowledge transfer. At the same time, Jayaraman *et al.* [44] propose an approach for zero-shot classification, for when image attributes are unreliable, and use the error tendencies of the different attributes to develop a linear discriminant model. There have been many such attribute based learning methods for visual recognition that have been developed in the past [45, 46, 47, 48, 49, 50], to point to a few.

Unfortunately, there are situations when such attribute information is not easily available, and therefore, we need to build a semantic space for labels. Building such

a space may lead to an additional overhead and it may also require an extensive knowledge of the domain in which labels are defined. For these reasons, the learned semantic embeddings may not even be of the required high quality. To address these concerns, a general methodology to learn word embeddings is presented by Mikolov *et al.* [40]. The technique, known as *word2vec*, has made a advances in both efficiency and quality of the vectors learned. These vector representations are easily available for billions of words trained on terabytes of *Google News* and *Wikipedia* data. By using semantic embeddings learned using word2vec, Norouzi *et al.* [51] proposed *ConSE*, a zero-shot image classification system specifically tested on image classification. ConSE uses a convolution network to embed an image into a vector space and uses a convex combination of nearest label embeddings to construct the prediction vector. It assumes that the set of predicted labels is disjoint to the set of seen labels, an assumption that is neither valid in the real world nor in our work. Another work [52] proposed a linear regression model for zero-shot classification that also relies on independently learned semantic embeddings. This work is very similar in practice to the model of Palatucci *et al.* [6]. Another work [53] proposed to learn semantic embeddings for labels from scratch without using independently learned semantic embeddings, which differentiates it from our work.

An interesting work based on semi-supervised learning [54] proposed a max-margin multi-class zero-shot classifier with the assumption that unlabeled data is available during training, an assumption that is not made in the paper defining the problem of zero-shot classification and that, therefore, we are not making in this chapter as well. One very recent work [5] proposes to learn semantic similarity embeddings for zero-shot classification, such that, each source or target label is represented as a mixture of seen label proportions. The method does well to predict unseen labels when the set of possible labels does not include seen labels, since the model is biased towards predicting seen labels. But, the method does not perform as well in the real world scenarios where it is not possible to distinguish between an unseen and a seen label at test time. Contrary to this, we work on a method that focuses on learning an accurate mapping from input data to the semantic space

representation of labels, which leads to better accuracy in predicting the correct label.

2.2 Proposed Method

In this section, we present our two step approach to zero-shot classification. We first describe the method to learn the proposed data-dependent embeddings that we refer to as *property embeddings*. Afterwards, we present how to minimize the classification error for nearest neighbour zero-shot multi-class classification using these property embeddings.

2.2.1 Property Embeddings

Most previous methods either assume the presence of reliable attribute information for labels [52] or directly use noisy word embeddings [51]. While some other methods make assumptions about the availability of unlabeled data during training [54], when in fact, this is often not possible in the real world. On the other hand, semantic embeddings like word2vec [40] and Glove [55], or more recently FastText [56] - while easy to obtain - are generally learned independently of the task at hand. Consequently, these embeddings are noisy, and therefore, can lead to inferior performance on the task at hand. Hence, we attempt to re-learn the label embeddings (obtained using word2vec on google news) *explicitly* using task data, and refer to them as property embeddings.

There is a one-to-one correspondence between labels and their property embeddings i.e. each label has a label embedding and a property embedding. These property embeddings encode labels such that: 1) the new property embeddings can be learnt from the input; 2) the similarities among labels in the label embeddings space are preserved in the property embedding space as much as possible. This leads us to formulate the objective J_s in Equation (2.1) in which the first part ensures that the matrix of property embeddings B can be mapped from data X using the model W, while the second part ensures that the cosine similarities that existed between different labels in the original label embeddings are preserved. We take the average of the features for all instances that have the same label, therefore, we have one input $\mathbf{x}$ for each label. We denote by $\mathscr{S}/\mathscr{U}$ the set of seen/unseen labels and the subscript

s/u refers to parameters corresponding to seen/unseen labels.

$$J_s = \alpha \|XW - B_s\|^2 + (1-\alpha)\|B_s B_s^T - L_s L_s^T\|^2 + \lambda \|W\|^2 \tag{2.1}$$

$$W, B_s = \arg\min_{W,B_s} J_s \tag{2.2}$$

Here $X \in \mathbb{R}^{|\mathscr{S}| \times d}$ is the input matrix, $B_s \in \mathbb{R}^{|\mathscr{S}| \times k_1}$ is the new property embedding matrix, $L_s \in \mathbb{R}^{|\mathscr{S}| \times k_2}$ is the label embedding matrix, $W \in \mathbb{R}^{d \times k_1}$ is a linear model, and d, k_1, k_2 are the dimensions of the input, property embedding and label embedding respectively. Please note that W is only used to learn B_s, once we obtain B_s we throw away the learned W, and re-learn its neighbourhood sensitive version in the next section. Please also note that the dimensions of property embeddings can be different from label embeddings.

We model a slightly different objective to learn property embeddings for zero-shot labels (i.e. B_u). The only piece of information we have about zero-shot labels is their relative similarity with other labels in the original label embedding space. This is because we do not observe these labels in the train set. Therefore, we need to preserve the similarity between these zero-shot labels and the seen labels in the new property embedding space, leading us to the objective:

$$J_u = \|B_s B_u^T - L_s L_u^T\|^2 \tag{2.3}$$

$$B_u = \arg\min_{B_u} J_u \tag{2.4}$$

Please note that B_s is fixed in this objective and we use the value obtained in Equation 2.2.

We optimize both objectives (Eq 2.2 and Eq 2.4) using gradient descent, but we recognize that Eq 2.4 can also be solved in closed form. We also acknowledge that there might be other methods that can be applied to solve these objectives e.g. second-order methods, which might differ from gradient descent in terms of time-space performance, but that should not affect the quality of the final results.

2.2.2 Neighborhood Sensitive Mapping

In this section, we propose an objective for nearest neighbor classification that takes into account the neighbourhood of the label while mapping input into the property embedding space. Our aim is to learn labels that are in crowded (i.e. have a lot of labels nearer to them in the property embedding space) neighbourhood with more precision at the expense of labels that are in sparse neighbourhood. This is because a label in crowded neighbourhood is more likely to be misclassified by a nearest neighbour classifier, as compared to a label in a sparse neighbourhood, therefore, we refer to the approach as *Neighborhood Sensitive Mapping*. To achieve this, we formulate the objective as:

$$W = \arg\max_{W} \sum_{(\mathbf{x},\hat{q})\in\mathscr{D}} \left(\sum_{\tilde{q}\in(\mathscr{S}\bigcup\mathscr{U})-\hat{q}} \log\left(\sigma\left(\mathbf{x}\,W\left(\mathbf{b}_{\hat{q}}-\mathbf{b}_{\tilde{q}}\right)^{T}\right)\right)\right) - \lambda\|W\|^2 \quad (2.5)$$

Here $\mathbf{x}\in\mathbb{R}^{1\times d}$ is the input, $\hat{q}$ is the true label, $\mathscr{S}$ is the set of seen labels, $\mathscr{U}$ is the set of unseen labels, $\mathbf{b}_q\in\mathbb{R}^{1\times k_1}$ is the property embedding of label q, σ is the sigmoid function and $W\in\mathbb{R}^{d\times k_1}$ is a linear mapping from input to property embedding space.

The first part of the above objective ensures that the dot product of the prediction vector ($\mathbf{x}W$) with the property embedding of the true label ($\mathbf{b}_{\hat{q}}$) is larger in comparison to the other labels. While the second part is the standard $l2$ regularization over the learned parameter W.

2.3 Experimentation

In this section, we first describe the fMRI dataset that was used for experimentation. After that, we pose three different research questions and analyze the results in light of these questions over the fMRI dataset, exactly as in the original zero-shot paper by Palatucci *et al.* [6], which also analyzed these three questions over fMRI dataset, and also draw comparisons with competitive baselines. After the analysis over these research questions, we describe two additional text classification datasets. We perform final experiments on these two text tagging datasets and discuss the obtained results.

2.3.1 fMRI Data

We used the fMRI dataset for experimental analysis over three research questions, exactly as in the original paper on zero-shot [6]. The fMRI dataset is composed of the neural activity observed from nine human participants while looking at 60 different concrete words. These 60 words are divided into 12 categories, like animals: bear, dog, cat, cow, horse and vehicles: truck, car, train, airplane, bicycle. Each participant was shown a word and a small line drawing of the concrete object the word represents. The participants were asked to think about the properties of these objects for several seconds while scans of their brain activity were recorded. Each sample measures the neural activity at roughly 20,000 locations in the brain. Six fMRI scans were taken for each word. We also used the same time-averaging described in Mitchell *et al.* [57, 6] to create a single average brain activity pattern for each of the 60 words, for each participant. The semantic embeddings used in the experiments were 300 dimensional vectors trained on Google news using *word2vec* [40], and are freely available online.

2.3.2 Research Questions

In this section we perform experimental analysis in the light of three different research questions that were posed in the original zero-shot paper [6], and compared our results to LM [6] and ConSE [51]. For this analysis, we use fMRI dataset as used by Palatucci *et al.* [6].

We refer to our proposed method as *NSM* i.e. Neighborhood Sensitive Mapping, Palatucci *et al.* [6] is referred to as *LM* and Norouzi *et al.* [51] is referred to as *ConSE*. In order to get insights into the effect of our novel property embeddings, we also test both NSM and LM with property embeddings. Therefore, *NSM-PB* and *LM-PB* refer to NSM using property embeddings and LM using property embeddings respectively, on the other hand, *NSM* and *LM* use the label embeddings.

1. How well can the model differentiate between two novel classes, where neither class appears in the train dataset?

We randomly draw a set of 1000 pairs of classes. We select one pair at a time and remove both of the classes in the pair from the training set, and then evaluate the

accuracy of binary classification between these two removed classes on the test set. We then report the average binary classification accuracy over these pairs.

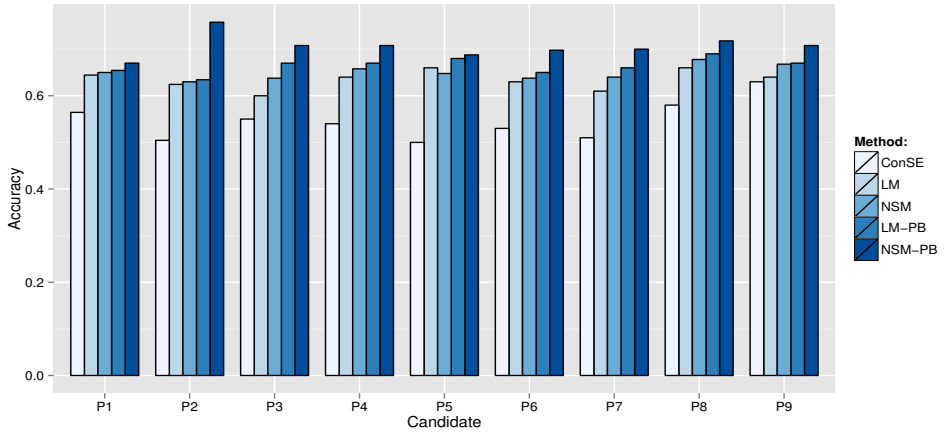


Figure 2.1: Binary classification accuracy for different participants in fMRI dataset using our proposed approach (referred to as **NSM-PB**) versus LM-PB, LM, NSM and ConSE. The results are averaged over 1000 different pairs of binary classes and the results are statistically significant at p-value = 0.0016 using a two sided t-test. The average results for NSM-PB, LM-PB, ConSE, LM and NSM: 0.7049, 0.6633, 0.5454, 0.6420 and 0.6495 respectively. Please note that the classification accuracy of a random classifier would be 0.5.

We can see that NSM-PB outperforms LM-PB, LM (Figure 2.1) and ConSE. The difference between NSM-PB and others is statistically significant using a t-test at p-value < 0.001. It can be seen that relearning of semantic embedding using the data, as well as modified neighborhood sensitive approach manages to better differentiate between novel classes. In fact, during our experimentation we observed that it works particularly well in case of classes that are very close in original semantic space, e.g. *hammer* and *chisel*. In those cases the use of a LM leads to results that are no better than random. We obtained an accuracy of 50% for *hammer* and *chisel* using LM, whereas 67% accuracy using NSM-PB.

2. How well can the model classify accurately in a multi-class classification setting, where all the test classes are absent in the train dataset?

We randomly select a set of 1000 groups such that each group consists of five different classes. We compute the average of the results of multi-class classification accuracy on all these groups. We restrict the predicted class to the classes in the group, i.e. we try to predict the correct class from among the five classes in each group.

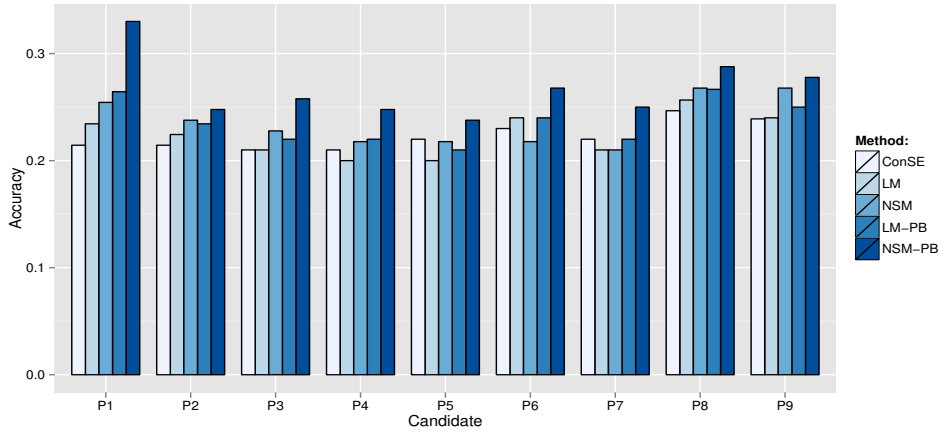


Figure 2.2: The figure plots multi-class classification accuracy for different participants in fMRI dataset. The figure compares our proposed approach (referred to as **NSM-PB**) against LM-PB, LM, NSM and ConSE. The results are averaged over 1000 different set of five classes and the results are statistically significant at p-value = 0.0015 using a two sided t-test. In this experiment, we test the ability of NSM-PB to distinguish between five novel classes that have not been seen in the train set. The average results for NSM-PB, LM-PB, ConSE, LM and NSM are: 0.2671, 0.2356, 0.2281, 0.22393 and 0.2349 respectively.

The results are presented in Figure 2.2 and are statistically significant using a two sided t-test at p-value = 0.0015. Even in a multi-class setting, property embedding based models outperform the semantic embedding based models. Also, NSM-PB outperforms ConSE, even though ConSE is competitive.

We can see in Figure 2.2 that NSM-PB performs better than both LM and ConSE. It shows that the proposed model is better at discriminating between multiple zero-shot classes in a multi-class classification setting. We can see in Figure 2.3 that the mean rank of the correct label in the prediction list is much lower in case of NSM-PB as compared to both LM-PB, LM and ConSE. It shows the effectiveness of NSM-PB in predicting correct labels higher in the prediction list.

3. How well can the model predict accurately in a multi-class classification setting, where the classifier has to choose from all possible classes?

For this task, we select a random class and remove it from the train dataset. Thereafter, we try to predict that class during testing from the set of all classes, including both novel and seen classes. It means that the classifier has to choose a class from among the 60 different classes present in the dataset. This is the hardest

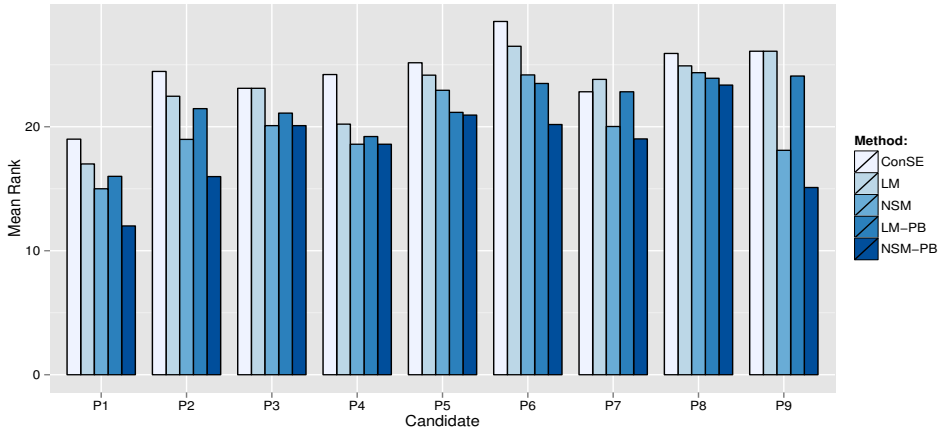


Figure 2.3: The figure plots mean rank for a true novel class (not seen in the train set) in the ordered prediction list of classes for different participants. In the figure our proposed approach (referred to as **NSM-PB**) is compared against LM-PB, LM, NSM and ConSE. The difference of NSM-PB against LM and ConSE is statistically significant at p-value < 0.001 using a two sided t-test. The average results for NSM-PB, LM-PB, ConSE, LM and NSM are: 18.58, 21.47, 24.36, 23.137 and 20.25 respectively.

task and also closely resembles the real world situation where we do not know in advance whether the instance belongs to a seen or unseen class. The results are presented in Figure 2.4 and are statistically significant using a t-test at p-value $<$ 0.001. NSM-PB outperforms the baselines in this task as well (see Figure 2.4). We make the classifier choose from the complete set of 60 classes instead of restricting the possible set of classes. Note that we outperform LM-PB, LM, NSM and ConSE by a significant margin in terms of classification accuracy (See Figure 2.4). These results are as expected given that we minimize generalization error for nearest neighbor classifier. This also clearly shows in the results as both LM and ConSE fall short in comparison to NSM for classification accuracy. These results clearly give us an insight regarding the usefulness of property embeddings, as LM-PB performs much better than LM and NSM-PB performs much better than NSM. In addition to property embeddings, the neighborhood sensitive mapping further improves the accuracy of classification.

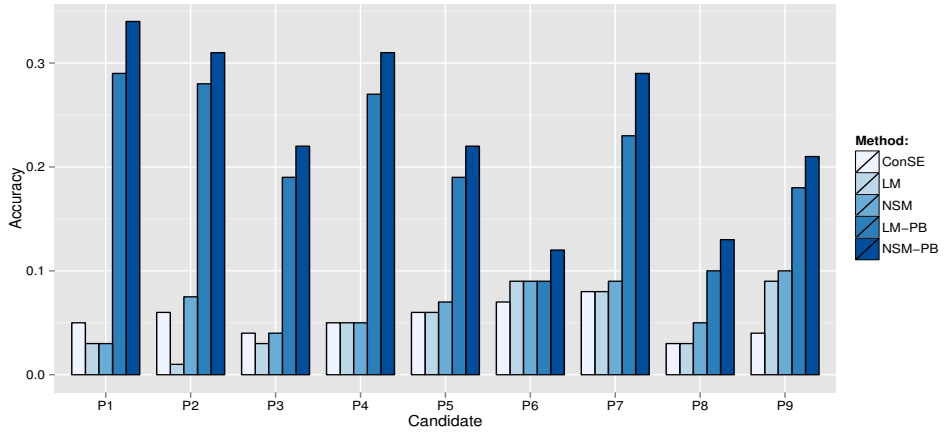


Figure 2.4: The figure plots mean accuracy for a true novel class (not seen in the train set) for different participants in fMRI dataset. In the figure our proposed approach (referred to as **NSM-PB**) is compared against LM-PB, LM, NSM and ConSE. The difference of NSM-PB against LM and ConSE is statistically significant at p-value < 0.001 using a two sided t-test. The average results for NSM-PB, LM-PB, ConSE, LM and NSM are: 0.2389, 0.2000, 0.0533, 0.0522 and 0.0661 respectively.

2.3.3 Text Classification Datasets

After the initial experiments on fMRI dataset, we tested the best performing NSM-PB against LM and ConSE on two larger text classification datasets. *First*, **wiki10+** dataset [58] contains text of *wikipedia* articles and the tags assigned by users on *delicious.com* for *url* of those articles. We used the most popular tag for an article as the label of that article. There were 20762 instances in the dataset with 5303 distinct labels. We cleaned the data of all html tags and computed tf-idf representations of the data. Afterwards, we used truncated-SVD to reduce noise in the data. *Second*, **delicious** dataset [59] contains features of web pages from all over the internet with tags generated by users on those web pages as the labels. The dataset contains 500 features for each instance and 983 unique labels. It has more than 16000 instances and the features are binary, with 1 indicating the presence of the feature and 0 indicating the absence of the feature.

2.3.4 Experiments

In this section we compare our approach against an additional baseline called SSE [5], which was a more recent method for zero-shot classification.

K	NSM-PB	LM	ConSE	SSE
5	**0.0685**	0.0261	0.0283	0.0269
10	**0.1344**	0.0392	0.0472	0.0481
50	**0.3590**	0.0521	0.0825	0.1541

Table 2.1: The value of accuracy for different methods on wiki10+ dataset. A given prediction is considered accurate if top-K labels in the prediction list contain the correct label. Results are statistically significant using t-test at p-value < 0.001.

K	NSM-PB	LM	ConSE	SSE
5	**0.0509**	0.0320	0.0433	0.0274
10	**0.0924**	0.0492	0.0822	0.0430
20	**0.1486**	0.0721	0.1350	0.0597

Table 2.2: The value of accuracy for different methods on delicious dataset. A given prediction is considered accurate if top-K labels in the prediction list contain the correct label. Results are statistically significant using t-test at p-value < 0.001.

We decided to test the accuracy of the methods using the top-K predicted labels, which means that the prediction was considered accurate if the top-K predicted labels contained the correct label. We can see in Table 2.1 that NSM-PB outperforms the other approaches by a considerable margin for varying levels of K. We can see that NSM-PB performed well even for $K = 5$, which is a very small value considering that the classifier has more than 5000 labels to choose from. We can see very similar results in Table 2.2 that NSM-PB again outperforms all other approaches for varying values of K.

2.3.5 Runtime

We observed that the running time for LM, ConSE and NSM were comparable, but the running time for SSE was much larger. The larger running time in the case of SSE can be attributed to sequentially learning label embeddings for all the labels in terms of proportions of seen labels. Therefore, SSE can take significant execution time when the seen label set contains a large number of labels.

2.3.6 Reproducibility

We share the codes used for the given experiments at `http://bit.ly/1RCNlwR`. The values of α and λ (Equation 2.1) were tuned over a validation set, and were

found to be 0.1 and 0.5 respectively. In case of ConSE, we use a multi-class logistic regression classifier for predicting class probabilities. The values of parameter T (i.e. number of top-T nearest embeddings for a given instance) in ConSE that gave best result was 5. The dimensionality of the learned property embeddings in the experiments was 10. The code for SSE was obtained from the matlab demo shared by the authors on their web-page.

2.4 Conclusions

We proposed a model that uses independently learned semantic embeddings to improve a zero-shot text classifier in a multi-class classification setting. We first describe the issues associated with zero-shot classification systems that use semantic embeddings. Then we highlight the shortcomings of classifiers that are not sensitive to the neighborhood of a label in the semantic space. After that we show that our proposed data-dependent property embeddings combined with neighbourhood sensitive mapping leads to improved results for zero-shot classification in text. These novel embeddings reduce noise in the pre-trained semantic embeddings, the proposed classification model then uses these newly learned embeddings for neighborhood-aware nearest neighbour classification.

Chapter 3

Improved Active Learning in Systematic Reviews

Systematic reviews are essential in various domains to summarize evidence from multiple sources. They are used in the formulation of public policies and also contribute in medicine in the development of clinical guidance [60, 61]. They involve searching, screening and synthesis of research evidence from multiple sources available in the form of published documents. It is critical in systematic reviews to identify all studies relevant to the review in order to minimise bias. It is easy to see that such a process can be extremely time-consuming and demand significant human resources [62]. Therefore, researchers have exploited active learning text classification to make the process of systematic review more efficient, which is an iterative process that starts from a small set of labelled studies and gradually learns to differentiate between relevant and irrelevant studies [16, 14, 63].

An essential component to the success of supervised learning is feature extraction. More recently, Hashimoto *et al.* [9] presented a topic detection model to improve the performance of the active learning classifier. Their method is based on a neural model - referred to as paragraph vectors [64] - to extract fixed length feature representations for the documents. To ascertain whether the proposed feature extraction model indeed works well, we experimented on a comprehensive dataset of reviews derived from diverse domains, and compared different feature extraction techniques with Hashimoto *et al.* [9]. Contrary to the published findings, we ob-

served that different feature extraction methods work well for different domains but none of them works well for all reviews.

After feature extraction, the other major component for systematic review classification is the active learning algorithm. Previously, naive active learning algorithms were used for this task with some success [14, 9], but a naive algorithm suffers from the problem of extracting studies that are similar to the previously retrieved studies. This is because the active learning classifier may not receive enough diverse samples initially, and then continues retrieving studies similar to the ones that have already been retrieved.

To address this, we propose a novel active learning algorithm for clinical text classification that reduces bias (i.e. 'bias' towards selecting studies similar to the ones chosen in the beginning) by including novelty as a criteria in addition to relevance. We should mention that diversity based active learning techniques have been proposed earlier [65, 66, 67], but to the best of our knowledge, they have not been applied for the task of citation screening.

At the end, we derive meaningful insights from our experiments and describe a process for choosing the best feature extraction model for any given review. Our contributions can be summarized as follows:

- We perform experiments on the use of different feature extraction models over a comprehensive set of reviews, and contrary to the published results in Hashimoto *et al.* [9], observe on a larger dataset that paragraph vector based topic modelling does not work better compared to bag-of-words and/or simple paragraph vectors based approach for active learning in systematic reviews.

- We propose a simple active learning algorithm that removes the inevitable bias in naive algorithms [13, 9] based solely on relevance, and instead uses both novelty and relevance during the initial phase.

- We derive insights from our experiments on the performance of different feature extraction models for different domains, and propose an effective way to choose the best feature extraction model that requires no prior knowledge

about the review.

3.1 Related Works

An interesting work [62] conducted a comprehensive survey of different machine learning methods used for advancing automation in systematic reviews. One more such work [68], described the applications of various text mining technologies for citation screening, these were namely, automatic term recognition, document clustering, classification and summarization; these techniques have been used in a number of previous works for the identification of relevant studies in systematic reviews. There are two major components of a citation screening system: 1) feature extraction model; and 2) active learning strategies. We will discuss related works on both of these aspects.

Wallace *et al.* [14] proposed a multi-view approach that encodes documents in terms of different feature vectors, such as, words that appear in the title and abstract, keywords and MeSH terms. Afterwards, each feature space is used to train a separate classifier, and an ensemble of those based on majority voting predicts the final outcome. Although, such previous feature extraction methods worked well for studies in the clinical domain, but it was noticed in Miwa *et al.* [16] that the task of identifying relevant studies in public health domain had poor performance. They argued that the task of identifying relevant studies is more challenging in this domain due to the presence of documents from wide ranging disciplines (e.g. social science, occupational health, education etc.) compared to more specific studies included in clinical systematic reviews [69]. To address this, the authors proposed using topic modelling to extract topic based features using Latent Dirichlet Allocation (LDA) [70], though, other topic models available in machine learning can be used just as well e.g. probabilistic latent semantic indexing.

Other works have focused on improving the process of active learning. Settles *et al.* [71] presented a comprehensive survey on these methods. A more recent work [72] proposed an active learning algorithm based using group of words that describes a label, known as rationals. These rationals were generated by a human

annotator and led to an extra manual effort. Unfortunately, it is often difficult to obtain additional annotations for the labels due to the expensive manual effort involved. A different work [14] proposed a semi-automated screening of biomedical citations for systematic reviews. They propose a novel online learning strategy that uses an ensemble of SVMs trained on different feature-spaces (e.g. title or abstract) to mark citations as "relevant" or "irrelevant". The studies that the classifier was least confident about were then sent to human experts for manual annotation. A number of works [65, 66, 67] have also attempted to add diversity in active learning process, but to the best of our knowledge, they have not been applied for the task of citation screening.

3.2 Methods

In this section, we briefly describe different feature extraction methods that have been used in previous works. We then describe the motivation and details of our proposed active learning algorithm.

3.2.1 Feature Extraction

- **Bag-of-words**: The most basic feature extraction models from text documents is called *bag-of-words*. In this model each of the possible terms in the entire corpus of documents is used to construct a vocabulary. At times the terms are weighted by tf-idf that increase (decreases) the relative weighing of unique (common) words in the document.

- **Latent Dirichlet Allocation**: Topic modelling techniques e.g. LDA [70], have been used in the area of systematic reviews to extract topical features from studies [73]. Such topical representations extracted from the documents are then used as features for training the active learning classifier.

- **Paragraph Vectors**: Recently, a neural network model has been proposed that learns word vectors and paragraph vectors (PV) in a joint manner [64], which is in contrast to the approach of learning them separately. Originally, word vectors represented only the words, but paragraph vectors are able to represent

a sequence of words in the form of phrases, sentences or paragraphs. Paragraph vectors have been reported to work with success in some previous works [9], and are expected to encode natural language better than other topic models. They have been used to measure similarities between *Wikipedia* articles and research papers [74]. They have also been used to extract topics from studies used in systematic reviews [9].

- **Topic Modelling using Paragraph Vectors**: Hashimoto *et al.* [9] proposed a technique for topic modelling using paragraph vectors. They argued that paragraph vectors lead to better feature extraction by jointly learning the vector representation of words and documents. Hence, clustering such paragraph vectors can lead to better topic assignments, in comparison to models based on bag-of-words representation e.g. LDA [70].

- **Clustering bag-of-words**: We can cluster the tf-idf based bag-of-words representation of documents to obtain topics. The documents can then be represented in terms of the cluster-distance matrix. It is the most basic of the different topic extraction models that was previously used as a baseline [9].

3.2.2 Active Learning

3.2.2.1 Background

The active learning process begins with a small number of manually labelled documents. These documents are then used to train a classifier that can be used to differentiate relevant and irrelevant studies among the rest of the studies. These studies are ordered in decreasing probability of relevance and the top-k studies are manually reviewed by an expert reviewer. These manually reviewed k studies are then used to retrain the active learning classifier along with the previously labelled studies.

There is an obvious problem with such a naive active learning algorithm. The classifier can easily get biased towards studies that are chosen in the beginning of the process. Such a classifier would continue to look for similar studies based on its current knowledge. In contrast, we hypothesized that including novelty in

addition to relevance while choosing documents for active learning can lead to an overall improvement in performance. Some other approaches like [72] require additional information about the labels, and can only work well with bag-of-words representation of documents. In comparison, our method can work well with both bag-of-words and distributed representations (e.g. paragraph vectors). To the best of our knowledge, naive active learning is the only algorithm used with success in systematic reviews [9, 62].

3.2.2.2 Proposed Algorithm

To solve the above mentioned problems, we extract topics from documents using LDA. It gives us output topic vectors for each document in the corpus $v(d) \in \mathbb{R}^{k\times 1}$. These topic vectors from different documents form a matrix $V \in \mathbb{R}^{n\times k}$, where n is the number of studies in the corpus and k is the number of topics. We create a separate matrix of topic vectors for documents d that have already been manually labelled. We refer to the set of already labelled documents as $\mathscr{H}$, and the matrix of topic vectors for documents $d \in \mathscr{H}$ is denoted as $S \in \mathbb{R}^{|\mathscr{H}|\times k}$. We denote the set of currently unlabelled documents as $\mathscr{G}$, and set of all documents as $\mathscr{D}$. We use principal component analysis to compute the top-t principal eigenvectors of $S^T S$. We arbitrarily define the probability of a document being novel as:

$$p(n|d) = 1 - \frac{\|UU^T v(d)\|_2}{\|v(d)\|_2} \tag{3.1}$$

where $U \in \mathbb{R}^{k\times t}$ contains t principal eigenvectors, and $UU^T \in \mathbb{R}^{k\times k}$ is the subspace formed by top-t principal eigenvectors of $S^T S$. It basically measures the novelty of a document by projecting its topic vector on the principal subspace formed by topic vectors of documents in the training set. The projection should be small for a document to be considered novel and vice versa. We should mention that we chose this way of computing novelty because of its simplicity and reduction of noise as we only consider top-t eigenvectors, but we recognise that there can be other ways of computing novelty as well e.g. distance from mean of the topic vectors. Additionally, we obtain the probability of a given document d being relevant $p(r|d)$ from the

classifier. We compute the probability of document being both relevant and novel as:

$$p(r,n|d) = p(r|d) * p(n|d) \tag{3.2}$$

which is under the assumption that novelty and relevance are independent given the document. We order the documents by the above mentioned $p(r,n|d)$, and then select the top-k documents in each iterative step of the active learning algorithm for review by an expert reviewer. We continue using novelty of a document in the active learning process till we have discovered a certain fixed number of topics. We assign a document d to topic i if

$$i = \arg\max_{k} \; v_k(d)$$

where $v_k(d)$ is the value in k_{th} index of topic vector $v(d)$. Afterwards, we stop incorporating the novelty in the active learning process and continue solely based on relevance. We assume that a given topic has been discovered if any document in the labelled set $\mathscr{H}$ is assigned to that topic. We provide a more formal description of the above mentioned process in Algorithm 1.

3.2.3 Evaluation

3.2.3.1 Evaluation Method

First, we evaluate the performance of different feature extraction methods. For this, we use a logistic regression based linear classifier during active learning. Prior to choosing logistic regression, we tried linear-SVM [9, 16], but observed that performance was very similar, therefore, we decided to use logistic regression classifier as it models the posterior probability of a study being relevant $p(y|d)$.

Second, we compare our proposed active learning algorithm against a naive active learning approach, and another competitive baseline. The active learning process starts with a small set of manually labelled studies. Features are extracted from this labelled set using different feature extraction models mentioned above. At each step of the iterative process a fixed set of studies are reviewed manually by an expert reviewer. We extract a sample of top-k studies at each iterative step for manual labelling from the ordered list of candidate studies [63, 62].

Algorithm 1 Systematic Reviews Learning

```
1: procedure ACTIVE LEARNING
2:     # Get initial manually labelled set
3:     ℋ = GETINITIALLABELLEDSET
4:     𝒢 = 𝒟 − ℋ
5:     t = 3
6:     max_topics = 150
7:     s = 25
8:     n = 0
9:     # Get topical representation of docs using LDA
10:    V = LDA(D, n_topic=300)
11:    while n <max_topics do
12:        clf = Classifier()
13:        clf = TRAINCLASSIFIER(clf, ℋ)
14:        R = GETRELEVANCESCORES(clf, 𝒢)
15:        N = GETNOVELTYSCORES(V, ℋ, 𝒢, t)
16:        for ∀d ∈ 𝒢 do
17:            scores(d) = R(d) ∗ N(d)
18:        end for
19:        # Get top s documents as per scores
20:        𝒢′ = GETTOPKDOCUMENTS(scores, s)
21:        ℋ = ℋ ∪ 𝒢′
22:        𝒢 = 𝒢 − 𝒢′
23:        # Topics discovered in labelled set
24:        n = GETNUMBEROFTOPICSDISCOVERED(V, ℋ)
25:    end while
26:    while |𝒢| > 0 do
27:        clf = Classifier()
28:        clf = TRAINCLASSIFIER(clf, ℋ)
29:        R = GETRELEVANCESCORES(clf, 𝒢)
30:        for ∀d ∈ 𝒢 do
31:            scores(d) = R(d)
32:        end for
33:        # Get top s documents as per scores
34:        𝒢′ = GETTOPKDOCUMENTS(scores, s)
35:        ℋ = ℋ ∪ 𝒢′
36:        𝒢 = 𝒢 − 𝒢′
37:    end while
38: end procedure
```

Third, we derive insights from our experiments that would help us choose the correct feature extraction model without prior knowledge of the reviews.

All our datasets are already labelled by expert reviewers. During training we simulated a human feedback active learning strategy similar to the previously published works [9, 16, 14].

3.2.3.2 Parameter Tuning

We tune the different parameters of paragraph vectors (PV) using cross validation. We keep the number of topics in LDA at 300, as in [9]. We experimented with dimensionality of paragraph vectors and found that using 300 dimensional document vectors performed better in general across different reviews. We also experimented with higher values of the dimension such as 1000 (as in [9]) but the results were did not improve. We tuned the regularization parameter for the linear classifier, but observed that the results are extremely stable across different values of the regularization parameter and used the value that gave the best results. The number of principal eigenvectors that we use to compute $p(n|d)$ are 3, i.e. $t = 3$ (line 42 in Algorithm 1). The value of parameter max_topics was set to 150 in line 6 of Algorithm 1. It denotes the number of topics to be explored before we stop using novelty as a criteria in our active learning algorithm. We used the value of $s = 25$ in the algorithm. A small value for s implies that the classifier would have to be re-learned too often, and larger value would lead to selecting too few relevant documents in the beginning, and as a result a decrease in the overall performance.

3.2.3.3 Evaluation Metric

We evaluate the performance of our active learning process using a metric called WSS@95, which stands for Work Saved over Sampling at 95% yield. It can be expressed as:

$$WSS@95 = (1 - burden) \mid yield \geq 95\% \tag{3.3}$$

$$yield = \frac{TP^M + TP^A}{TP^M + TP^A + FN^A} \tag{3.4}$$

$$burden = \frac{TP^M + TN^M + TP^A + FP^A}{N} \tag{3.5}$$

where N are the # of studies and superscript M and A denote manual and automatic screening decisions. TP, FP, TN and FN stand for # of true positives, false positives, true negatives and false negatives respectively. These notations are the same as those mentioned in Hashimoto *et al.* [9]. We decided to use the WSS metric as it has been used in previously published works [9] on citation screening in systematic reviews.

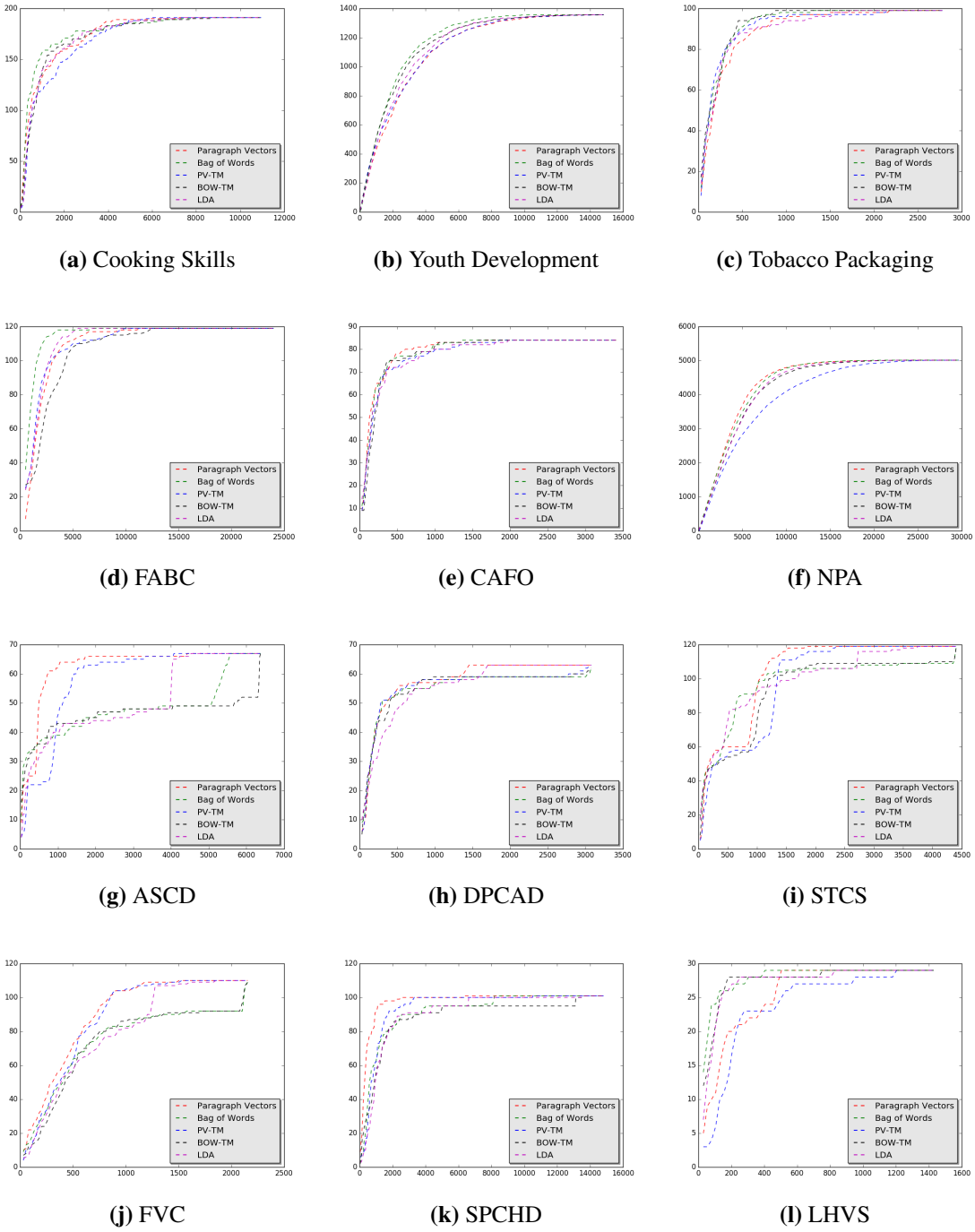
Dataset	Domain	Num. of citations	Fraction of relevant studies
LHVS	Clinical	1430	0.018
ASCD	Clinical	6381	0.010
FABC	Clinical	24469	0.005
DPCAD	Clinical	3087	0.019
STCS	Clinical	4415	0.026
FVC	Clinical	2157	0.050
SPCHD	Clinical	14841	0.006
NPA	Animal Studies	29659	0.168
CAFO	Animal Studies	3434	0.023
Cooking Skills	Public Health	10957	0.017
Youth Development	Public Health	14834	0.091
Tobacco Packaging	Public Health	2792	0.034

Table 3.1: Statistics of the systematic review datasets used in the experiment.

3.2.4 Datasets

We used a number of public health, animal study and clinical review datasets from completed systematic reviews. Some of these datasets have been previously used in Miwa *et al.* [16] and Hashimoto *et al.* [9]. We summarize the characteristics of different datasets used in our experiments in Table 3.1.

Brief Dataset Description. LHVS: Leukodepletion for patients undergoing heart valve surgery; ASCD: Amiodarone versus other pharmacological interventions for prevention of sudden cardiac death; FABC: Altering availability and proximity of products for changing selection and consumption of food, alcohol and tobacco; NPA: Animal studies on neuropathic pain; CAFO: Concentrated animal feeding operations; DPCAD: Psychological and pharmacological interventions for depression in patients with coronary artery disease; STCS: Preoperative statin therapy for patients undergoing cardiac surgery; FVC: Interventions for increasing fruit and vegetable consumption in children aged 5 years and under; SPCHD: Internet-based interventions for the secondary prevention of coronary heart disease; Youth Development, Cooking Skills and Plain Tobacco Packaging as described in [9, 16]



(a) Cooking Skills **(b)** Youth Development **(c)** Tobacco Packaging

(d) FABC **(e)** CAFO **(f)** NPA

(g) ASCD **(h)** DPCAD **(i)** STCS

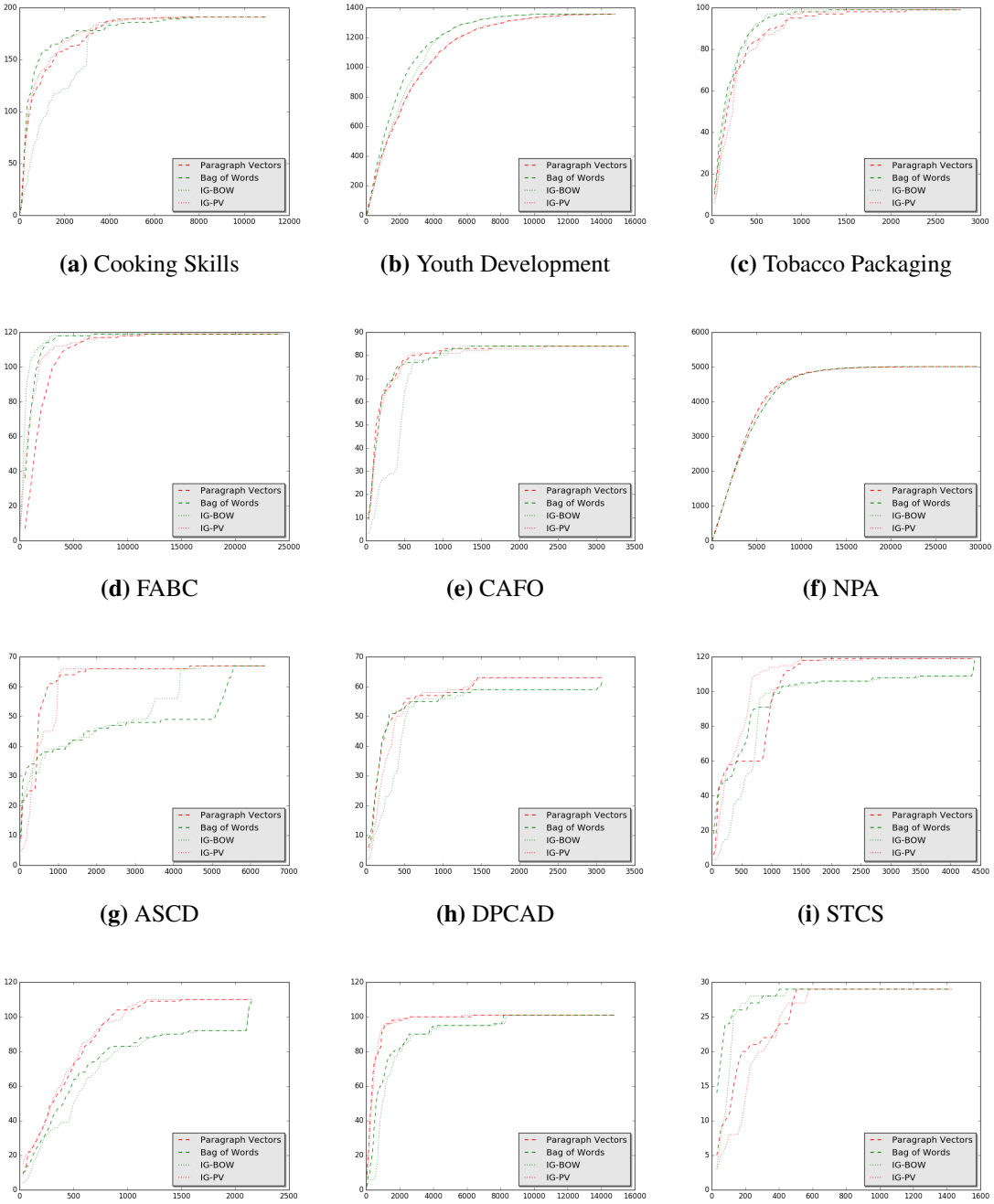
(j) FVC **(k)** SPCHD **(l)** LHVS

Figure 3.1: X-axis represents the number of documents that have been manually annotated and Y-axis represents the number of relevant documents that have been discovered. We can observe that different feature extraction methods work better for different reviews. BoW works better for public health documents and PV performs better on clinical studies. We can clearly infer that no feature extraction method is obviously superior to others over all documents.

3.3 Results

We investigate the performance of different feature extraction techniques in terms of relevant studies discovered and amount of manual annotations required. We plot the number of relevant studies identified for a given amount of manual annotation for different reviews that range from public health to clinical science. Both these quantities converge to their maximum value during the last iteration of the active learning process. We plot the performance for different feature extraction models in Figure 3.1. We denote topic modelling based on paragraph vectors with PV-TM, BoW stands for bag-of-words, BoW-TM denotes topic modelling based on BoW model and LDA refers to Latent Dirichlet Allocation. We can see in Figure 3.1 that different feature extraction methods work well for different studies. We observe that in majority of the cases, paragraph vectors and BoW models perform better than the rest. In documents pertaining to public health, BoW performs well, whereas for clinical reviews PV performs well. These results are contrary to previous results in [9], where the topic modelling based on PVs seemed to outperform other models across all disciplines. We observe that BoW and PV perform better than topic modelling based on paragraph vectors on most datasets. We show that it is not possible to find a single feature extraction method that performs superior to others across all domains, but we need to identify one per each review.

When we use our active learning algorithm with BoW model, then we refer to it as IG-BOW (Information Gain-BOW), whereas when we use the paragraph vector model, we refer to it as IG-PV (Information Gain-PV). We can observe the performance of proposed active learning algorithm compared to the naive active learning algorithm as the screening progresses in Figure 3.2. We can notice that our proposed algorithm explores in the initial phases of the screening process, but as the process continues the performance improves. It outperforms naive active learning algorithm by the end of the screening process. We plot the WSS@95 (i.e. WSS at 95% yield) in Figure 3.3 for our proposed active learning algorithm. In addition to the naive active learning algorithm, we compare our method with an additional **uncertainty**-based sampling algorithm that selects the samples about which the



(a) Cooking Skills **(b)** Youth Development **(c)** Tobacco Packaging

(d) FABC **(e)** CAFO **(f)** NPA

(g) ASCD **(h)** DPCAD **(i)** STCS

(j) FVC **(k)** SPCHD **(l)** LHVS

Figure 3.2: X-axis represents the number of documents that have been manually annotated and Y-axis represents the number of relevant documents that have been discovered. We can observe that our proposed active learning algorithm, that is IG-PV and IG-BOW explore during the initial phases of screening and then their performance improves as the process continues. We show the results of proposed active learning algorithm compared to the naive active learning algorithm in terms of WSS@95 in Figure 3.3.

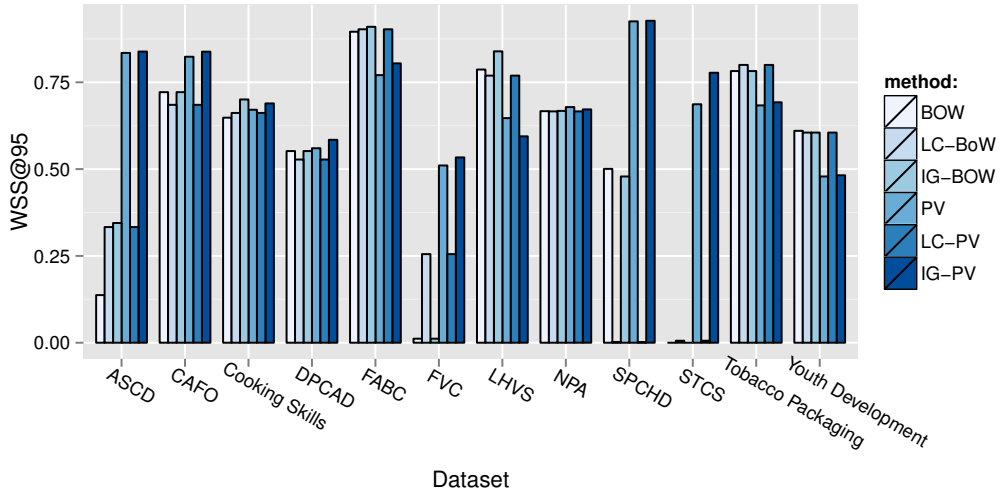


Figure 3.3: The figure plots the performance of proposed active learning algorithm and the naive active learning algorithm in terms of WSS@95. We use the two best performing feature extraction models (i.e. BoW and PV) for comparing the proposed active learning algorithm with the naive algorithm and baseline algorithm (LC-Bow/LC-PV). We can observe that in 9 out of the 12 datasets the proposed approach (IG-PV or IG-BoW) is the best performing in terms of WSS@95. The winners have a statistically significant lead over all the losers using a t-test at $p < 0.05$.

classifier is least confident during the initial 10% (set using cross validation on an independent validation set) screening i.e. it randomly selects k studies that have $0.4 \leq p(y = 1|d) \leq 0.6$. We refer to this baseline as LC, and consequently, LC-BoW uses bag-of-words as features and LC-PV uses paragraph vectors as features. We can observe that one of the two, i.e. IG-BoW or IG-PV, performs better compared to naive active learning algorithm using just BoW or PV. It validates our hypothesis that using novelty to explore during the initial phases of active learning can lead to better results overall, especially in terms of WSS@95. We only present BoW and PV because (as we mentioned in the previous paragraph) we observed that these two feature extraction methods perform well across most studies. We should mention at this point that we also used our active learning algorithm with the feature extraction model presented in [9], but it did not perform comparably to either PV or BoW. Therefore, we did not present the results with PV-TM to keep the analysis simple and sequential.

It has been shown in our results that both PV and BoW work well for specific

systematic reviews. It is not feasible to estimate with certainty in advance which of the two would work better for a review. Meanwhile, we obtained the recall for both feature extraction methods after screening through initial 10% of the data. We can see that comparison in Table 3.2. We report the recall for the initial 10% of the data and denote in bold the approach that scores higher in terms of WSS@95 by the end of the process. We observe that the approach that has higher recall on the initial 10% of the data works well in terms of WSS@95 by the end of the screening process. As a result, we infer that both approaches - IG-BOW or IG-PV - should be used in the beginning and then the one that performs better should be continued.

We should mention that we experimented with different ensembles of feature extraction models, but since the results were not impressive compared to the individual models, we omit the results.

Dataset	IG-BOW	IG-PV
LHVS	**0.896**	0.275
ASCD	0.537	**0.671**
FABC	**0.957**	0.899
NPA	0.458	**0.471**
CAFO	0.345	**0.821**
DPCAD	0.403	**0.634**
STCS	0.322	**0.613**
FVC	0.275	**0.345**
SPCHD	0.693	**0.950**
Cooking Skills	**0.497**	0.738
Youth Development	**0.435**	0.426
Tobacco Packaging	**0.757**	0.686

Table 3.2: Recall upon screening through 10% of the studies in the review. We use the bold notation to mark the overall winner in terms of WSS@95 at the end of the screening process.

3.4 Discussion

We should mention at this point that we also experimented with learning paragraph vectors using additional external data in the case of public health and animal studies. We expected that additional documents might lead to improved paragraph vectors. These external data consisted of studies related to the review in question, but most

of these were not directly relevant to the review. To our surprise, we did not see any improvement in the results upon using such external data for learning paragraph vectors. On the contrary, the performance decreased as we used more external data during the learning of paragraph vectors. We should also mention that we conducted our experiments on more than 30 reviews, and obtained results similar to those presented here.

We observed that active learning can be an effective strategy to semi-automate manual annotations and reduce the workload. Sadly, current active learning approaches do not accurately estimate the proportion of relevant studies that have been annotated. In many cases, it is necessary to extract 95% or more relevant studies in order to avoid bias in the systematic review. If there was an effective way to estimate the recall of the active learner precisely then the screening process could move towards complete automation. In the future, we will work towards an accurate recall estimation while using an efficient active learning strategy.

3.5 Conclusion

We evaluated different feature extraction models over a comprehensive dataset of reviews from varied domains. We observed that both BoW and PV can outperform other approaches over most reviews/domains. This led to the decision by Institute of Education at University College London to avoid overhauling their feature extraction system in favour of a newly published but unimpressive feature extraction model.

We recognized that a naive active learning algorithm suffers from bias. It tries to select documents that are similar to the documents in the initial training data. A small initial training set could then lead to reduced performance. To address this issue, we propose a novelty based active learning algorithm that works on exploring different topics during the initial phases and then proceeds based on relevance in the later phases. It leads to more exploration of different topics in the beginning and better performance in terms of WSS@95 by the end. We evaluate our approach against naive active learning algorithm, and observe that the proposed algorithm works as well as, or better than, the naive algorithm.

We also developed insights regarding the choice of feature extraction methods for different reviews. We observed BoW and PV to be the best feature extraction models over a large number of reviews, and inferred that both the feature extraction methods should be used during the screening of initial 10% studies. Afterwards, the better performing approach should be continued, which does lead to some extra manual annotations in the beginning, but the overall gain in terms of WSS@95 compensates for that.

Chapter 4

Automatic Structured Clinical Text Annotation

There has been rapid growth in the volume and diversity of available healthcare data, ranging from electronic health records (EHRs) to biomedical literature. This proliferation of data provides unprecedented opportunity to improve patient care [75, 76, 77, 78], but simultaneously the volume of published information makes it difficult to efficiently retrieve and compile relevant evidence. In this work we focus on biomedical literature, and in particular on texts that describe the conduct and results of randomized controlled trials (RCTs), which are considered the gold standard in evidence for particular interventions.

In general, the clinically salient aspects of an RCT include: (1) the Population(s) enrolled; (2) the Intervention and Comparator treatments administered (the distinction between these is arbitrary, and so these may be grouped); (3) the Outcomes measured. Collectively these are referred to as PICO elements. Clinical questions

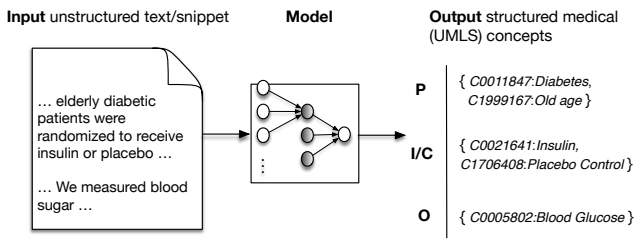


Figure 4.1: Illustration of the annotation task. The output comprises concepts drawn from the UMLS controlled medical vocabularly, grouped into terms that describe the study Population, Interventions/Comparators and Outcomes.

are widely considered answerable only when mapped onto a PICO frame. However, retrieving all articles that describe trials relevant to a given PICO frame (and hence question) is non-trivial, in part because reports of RCTs are communicated in unstructured (free-text) articles. Structured representations of articles that explicitly assign ontological terms to distinct PICO elements would support automated retrieval and question-answering systems [79]. We therefore aim to develop an automated approach to mapping from free-texts to distinct sets of terms from the Unified Medical Language System (UMLS) corresponding to each PICO element. This is depicted schematically in Figure 4.1.

This multilabel and multitask setting presents formidable challenges from a machine learning perspective. In particular, the output space is vast: there are more than 300K terms in the controlled medical vocabulary we are targeting (UMLS). Second, as is the case in many biomedical tasks, we have a relative dearth of available training data with which to estimate model parameters. Third, outputs (i.e., sets of UMLS terms corresponding to the respective PICO elements) are correlated: a given study population constrains the space of plausible interventions and outcomes. For example, if the population comprises *adult males*, it is unlikely that the outcome will be *time to labor induction*. These correlations between label outputs should be exploited. We address these problems in this chapter by introducing a novel neural approach involving two parts: *candidate term generation* and *selection/classification*.

The specific contribution of this work is a novel method for multilabel classification into multiple distinct, but correlated label sets using a neural model that considers 'candidate' label tuples, conditioned on the text being annotated. Our approach addresses training data sparsity by re-framing the annotation task as a two step process in which we first generate a set of candidate annotations relevant to the input text, and then we select and group these. In our case, we generate candidates using both (a) a multitask model directly trained to generate candidate concepts, and, (b) the *MetaMap* tool[1] [80]. We then use a neural discriminative model to infer plausible triplets of concepts from the unstructured candidate set, conditioned on the

[1] `https://metamap.nlm.nih.gov/`

free-text being annotated. We demonstrate that this model improves performance (compared to relevant baselines) on the important task of automatically annotating biomedical literature with structured UMLS concepts. As far as we aware, this is the first work to tackle this challenging problem.

While our motivating application concerns biomedical literature processing, we emphasize that the problem we consider is general, and the candidate-generator/discriminator approach we propose may have broad application for similarly structured tasks.

4.1 Related Work

We briefly review two threads of work related to our present effort: research on automated biomedical text annotation (Section 4.1.1) and then approaches to structured and multilabel classification. (Section 4.1.2).

4.1.1 Biomedical Text Annotation

Biomedical natural language processing is a broad, active field [81, 82]. Here we briefly review work relevant to our specific task of annotating text with structured PICO element concepts. One early system developed to extract clinical trial characteristics from free-texts is ExaCT [83], which aimed to identify and extract data elements from free texts describing clinical trials necessary for evidence synthesis. ExaCT used a hybrid of statistical and rule-based approaches. A similar system was developed by Summerscales [84]. Their system attempted to automatically calculate summary statistics reported in an abstract by first identifying treatment group and outcome mentions and then processing numerical quantities in the text with reference to these.

Related work has attempted to identify spans or sentences of texts describing trials that correspond to the PICO elements. For example, Boudin et al. [79] described ensemble methods for identifying sentences in abstracts corresponding to each PICO element [85]; they demonstrated that automatic PICO tagging can improve clinical IR. More recently, Wallace and colleagues developed a model of extracting PICO sentences from full-texts, by exploiting a novel form of distant

supervision [86].

Work has also been done on automatically assessing the 'risks of bias' in clinical trials, e.g., due to improper randomization, based on the text in the articles describing them. This work has entailed jointly extracting the sentences supporting these assessments [87, 88, 89, 90].

There is an annual BioASQ[2] challenge that involves applying MeSH terms to biomedical abstracts - discussed in the next chapter - but it differs from PICO tagging as it does not require applying labels corresponding to PICO aspects. As far as we are aware, the present work is the first to consider the task of mapping from free-texts to structured concepts explicitly corresponding to the respective PICO elements.

4.1.2 Structured Multilabel Classification

The task we have considered may be viewed as an instance of structured multilabel classification. There is of course a rich body of work on general multilabel classification (e.g., [91, 92, 93]). It is challenging to learn an accurate and effective multilabel classifier in domains with many labels [59, 94]. Label space reduction methods provide one means of mitigating the problem of large label spaces [95, 96, 97].

More specific to the current application, multilabel classification for text has also received a fair amount of attention [36, 98]. A classic approach for multilabel text classification is to posit a generative mixture model wherein documents are associated with a set of labels that are in turn ascribed partial responsibility for generating the words comprising a given document [98]. It is not clear how to generalize this approach to our setting, however, because: (1) Labels are *grouped* as PICO elements which implies a correlation between these label sets, i.e., documents are not associated with unstructured bags of labels; (2) Our output space (defined by a medical ontology) is vast, and thus a mixture model would require an unwieldy number of latent components.

Another sub-area of machine learning research relevant to our setting is multi-task learning [99]. In particular, the PICO elements (and associated multilabel sets) may be viewed as distinctive 'tasks'; thus we find ourselves in effectively a multitask

[2] http://bioasq.org/

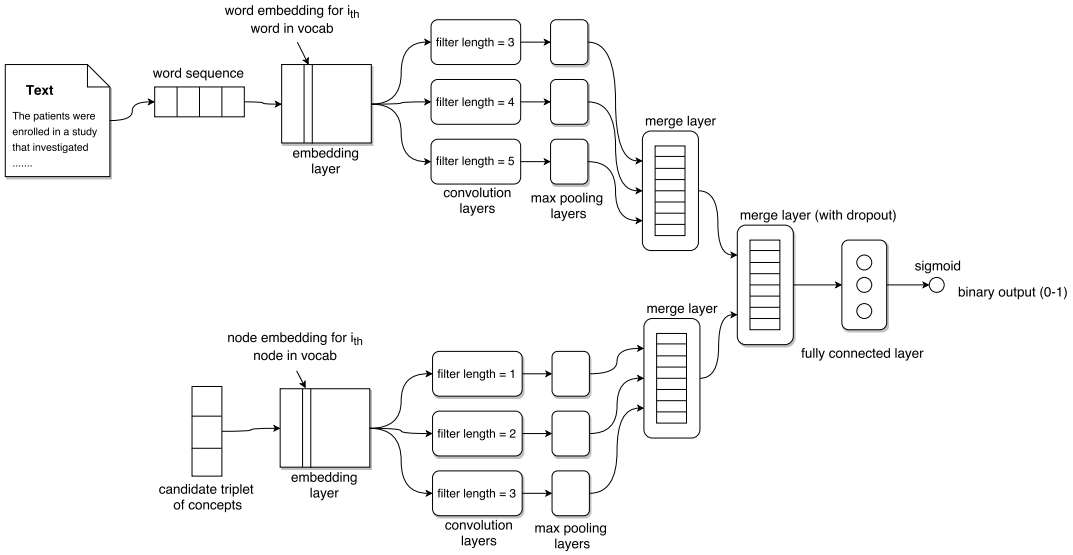


Figure 4.2: A schematic of our selector network variant *CS-joint*. This accepts as input the text snippets describing a study and a triplet of candidate concepts $(c_{\mathrm{P}}, c_{\mathrm{I/C}}, c_{\mathrm{O}})$, thus associating each candidate concept in the tuple with a particular PICO element. This induces a joint model that considers the likelihood of these three candidate concepts mapping to particular PICO concepts, given the input text. The output is a binary decision regarding the applicability of a candidate triplet.

multilabel setting. Standard multitask learning has been studied at length in general, and in the context of natural language processing in particular [100, 101]. Indeed, we build upon the basic neural multitask architecture in [100] as a component in our approach.

To the best of our knowledge, this is the first work to explicitly consider the problem of jointly annotating texts with ontological labels for multiple, correlated aspects.

4.2 Methods

Our proposed approach comprises two components. The first is a *candidate generator*, responsible for inducing an unstructured set of 'candidate' UMLS concepts deemed likely to apply to a given input text. Ideally this would be a high-recall (but possibly low-precision) set of terms. The second component is a *selector*, which accepts the candidate concepts as input, along with the text to be annotated, and conditioned on these selects and outputs likely structured sets of concepts, i.e., concepts pertaining to the aforementioned PICO elements.

Formally, denote an input text by $\mathbf{x}$. Then we run this through our candidate generator, g:

$$\mathscr{C} = g(\mathbf{x}) \tag{4.1}$$

and the outputs are consumed by the selector s:

$$\mathscr{Y} = s(\mathscr{C}, \mathbf{x}) = s(g(\mathbf{x})). \tag{4.2}$$

Here $\mathscr{Y}$ is assumed to be structured, i.e., include particular concepts corresponding to the PICO elements. Thus $\mathscr{Y} = \{\mathscr{Y}_{\text{P}}, \mathscr{Y}_{\text{I/C}}, \mathscr{Y}_{\text{O}}\}$.

This component approach affords the important advantage of allowing g to effectively map from the vast universe of possible structured terms (here, UMLS terms) to a relatively small set of those deemed reasonably likely for the text at hand. The selector model s can then perform more in-depth processing of candidates to infer likely configurations of candidate terms across the $\{$P, I/C, O$\}$ elements, taking into consideration correlations between these subsets. In our case, this architecture was motivated in part by the existence of MetaMap [10], a tool that uses rules and heuristics to map from free text snippets to possible terms. This forms one part of our generator model, complementing a purely data-driven approach.

4.2.1 Selector Model

We begin by describing the selector model, s, which assigns a subset of the concepts contained in an unstructured candidate set $\mathscr{C}$ to the respective PICO elements, conditioned on the input text. An instance of this model (described in greater detail below) is depicted in Figure 4.2. Following [32, 102], we adopt a convolutional neural network (CNN) to encode texts. Concretely, we accept input texts to be annotated as sequences of words that are passed to an embedding layer that associates vectors (distributed representations) with words, thus forming an input matrix. We initialize word embeddings to pre-trained vectors induced over the entire set of abstracts indexed on MEDLINE, a repository of biomedical literature; we update these representations during model training via back-propagation. We apply independent convolutional filters of varying length over this matrix. That is, these filters consume

one or more consecutive word embedding inputs at a time. Outputs from each filter are passed through a *max-pooling* operation to extract one scalar per filter (note that we use multiple filters of each filter size). These scalars are concatenated to form a final vector representation of the input text with a number of dimensions equal to the total number of convolutional filters. We will denote this induced representation of input i by $\mathbf{x}_i$.

The input text encoding approach just described is the same across the different Candidate-Selector (CS) model variants we discuss. What differs between them is the handling of the candidate concept(s) under consideration. For all variants we use embedded representations of controlled (UMLS) terms. We initialize these to pre-trained embeddings induced via *DeepWalk* [2], an approach to unsupervised distributed representation learning for graph-structured entities. During candidate classification, embedded representations of one or more candidate concepts are considered and the task is to decide whether these apply to the text under consideration, and if so, which PICO element they describe. We next describe three variants of our candidate-selector architecture, in ascending order of complexity.

CS-ind. The simplest variant of our model treats predictions regarding the designation of individual terms to respective PICO elements as independent, given the text. This model variant thus comprises three independent instances of the same model (i.e., with separate sets of parameters), one per PICO element. We concatenate the induced vector representations of the input text i and the (single) candidate concept under consideration (indexed by j) and estimate the probability of it being applicable to a given PICO element by running it through a logistic function σ:

$$\begin{aligned} P_e(\text{concept } j|\mathbf{x}_i^{(e)}, \mathbf{w}_o^{(e)}, \mathbf{C}^{(e)}, \mathbf{W}_h^{(e)}) &= \sigma(\mathbf{w}_o^{(e)} \cdot \mathbf{v}_h) \\ \mathbf{v}_h &= \lambda(\mathbf{W}_h^{(e)}[\mathbf{x}_i^{(e)} \oplus \mathbf{C}_j^{(e)}]) \end{aligned} \tag{4.3}$$

Where e indexes PICO elements and hence models, making explicit the fact that the respective PICO model parameter sets are independent; $\mathbf{x}_i^{(e)}$ denotes a vector representation of input text i (induced via a CNN); $\mathbf{w}_o^{(e)}$ a weight vector parame-

terizing the output probability model; $\mathbf{C}^{(e)}$ the concept embeddings matrix; $\mathbf{W}_h^{(e)}$ a weight matrix for a hidden dense layer; and $\oplus$ denotes vector concatenation. Here $\lambda(\cdot)$ denotes an element-wise activation function (in our case, identity) and dropout regularization [103]. We reiterate that these predictions are made separately for each PICO element.

CS-cond. The patient population enrolled in a trial is not independent of the interventions and outcomes considered, as the former will clearly influence the latter. A first attempt to exploit such correlations is our *CS-cond* model, which starts by predicting which candidate terms describe the population, and then conditions the subsequent selection of terms corresponding to interventions on these. Finally, the selection of outcomes terms is explicitly conditioned on the preceding two sets of terms (i.e., the terms designated as describing the study population and interventions).

More formally, we use *CS-ind* to select terms $\hat{\mathscr{Y}}_{\text{P}} \subseteq \mathscr{C}$. We then use a modified architecture for the models that select intervention and outcomes terms. In particular, the model for predicting interventions accepts a third input matrix comprising the stacked embeddings corresponding to the terms in $\hat{\mathscr{Y}}_{\text{P}}$.[3] Because the order of these terms is arbitrary, we pass only length 1 convolutional filters over this matrix (such filters consider a single concept at a time). We again apply max-pooling over these to induce a vector representations of the population concepts selected by the model in the preceding step, which we designate by $\mathbf{z}_i^{(P)}$.

$$\begin{aligned} P_{\text{I/C}}(\text{concept } j|\mathbf{x}_i^{(\text{I/C})}, \mathbf{w}_o^{(\text{I/C})}, \mathbf{C}^{(\text{I/C})}, \mathbf{W}_h^{(\text{I/C})}, \mathbf{z}_i^{(P)}) &= \sigma(\mathbf{w}_o^{(\text{I/C})} \cdot \mathbf{v}_h) \\ \mathbf{v}_h &= \lambda(\mathbf{W}_h^{(\text{I/C})}[\mathbf{x}_i^{(\text{I/C})} \oplus \mathbf{z}_i^{(P)} \oplus \mathbf{C}_j^{(\text{I/C})}]) \end{aligned} \tag{4.4}$$

The model for outcomes is analogous, except that it takes as an additional input a matrix comprising the embeddings for the terms selected both for populations and interventions/comparators, i.e., in addition to merging $\mathbf{z}_i^{(P)}$ to the model input we concatenate $\mathbf{z}_i^{(I/C)}$ before passing through the network. Thus the selection of outcomes terms is conditioned jointly on the inferred population and intervention descriptors.

[3] Operationally, we impose an upper-bound k on the number of terms that can be selected for a given element; thus the input matrix here is $k \times d$, where d is the embedding dimension.

CS-joint. Our final variant is a fully joint approach to selecting P, I/C and O candidate terms. This model consumes structured triplets as input (i.e., one candidate concept per PICO element) and estimates the conditional probability that these jointly apply to the text under consideration. The model is depicted schematically in Figure 4.2. In brief, we create an input matrix comprising the embeddings of the candidates in a given triplet, and run convolutional filters of lengths ranging from 1 to 3 over this input; this induces a vector representation of the triplet of candidate concepts which is then concatenated with the inferred representation of the input text to form a penultimate representation used to make a joint prediction concerning the applicability of the structured triplet of terms.

This model is attractive in that it affords a truly joint estimate regarding assignment of terms to PICO elements. However, it does mean that at test time we have to generate all combinations of candidate concepts to make predictions for possible triplets in turn.

4.2.2 Candidate Generation

Having presented our approaches for candidate selection s, we now turn our attention to *generating* candidates provided an input text, i.e., specification of g. Broadly, we consider two approaches here, the outputs of which we compose: in the first we use a separate, pre-existing system called MetaMap to generate an unstructured set of candidate terms. We also adopt a data-driven *learned* approach to candidate generation. We describe these in turn below.

4.2.2.1 MetaMap

MetaMap [80] is a tool developed by the National Library of Medicine (NLM) that assigns concepts from Unified Medical Language System (UMLS) vocabularies to free-texts. Note, however, that it does not attempt to categorize these assigned concepts into PICO elements. The UMLS is a meta-ontology, incorporating $\sim$200 standardised medical vocabularies. Synonymous terms are linked across vocabularies by unique semantic identifiers. MetaMap provides rich semantic information for biomedical informatics, but for our purposes it suffices to know that it implements a service which provides UMLS terms that match a given input text. We thus use

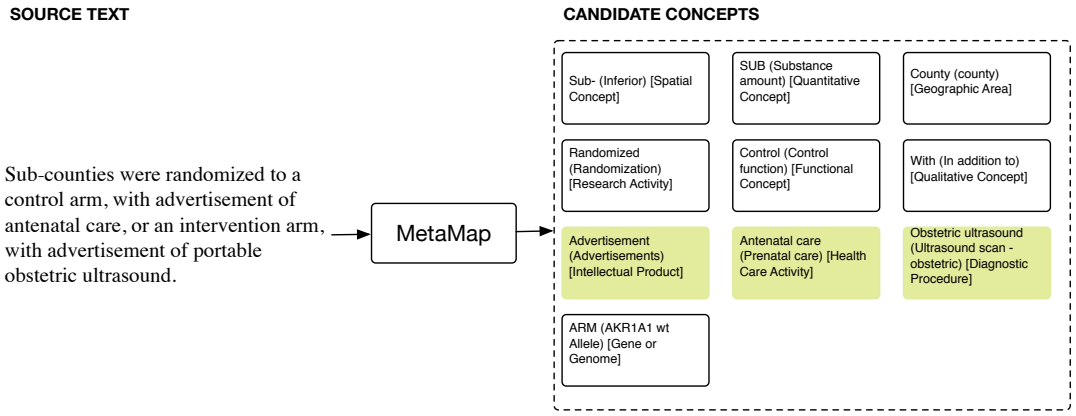


Figure 4.3: Schematic illustration of the use of MetaMap to generate a high recall set of candidate concepts. The target subset of concepts (here being those describing the interventions studied) are highlighted. Note that MetaMap output includes two types of noise: 1) An ambiguous string being assigned to the incorrect concept (e.g. 'Sub' being mapped to 'substance amount') and 2) the concept being correctly mapped from text but not describing our aspect of interest.

MetaMap to generate an initial list of unstructured candidate concepts. A schematic of this process is shown in Figure 4.3. In general, under the settings used here, we found the candidate set generated by MetaMap to be high recall but relatively low precision.

4.2.2.2 Learning to Generate Candidates

In addition to MetaMap, we consider the approach of directly predicting UMLS concepts corresponding to the respective PICO elements from free-text. This model is one of the baseline approaches to which we compare our proposed Candidate-Selector models. Learning to map directly from free-text to structured UMLS terms has the advantage of allowing recognition of concepts not identified by MetaMap (the recall of the generated candidate set is an upper bound on the recall the selector model will be able to achieve). However, the disadvantage of this approach is that the output space is vast: there are hundreds of thousands of concepts; learning to predict directly into this space is thus challenging, especially given our limited training data. Additionally, as we discuss further below, this approach precludes the possibility of identifying concepts that were not encountered during model training.

To directly predict candidate terms for input texts we adopt a convolutional

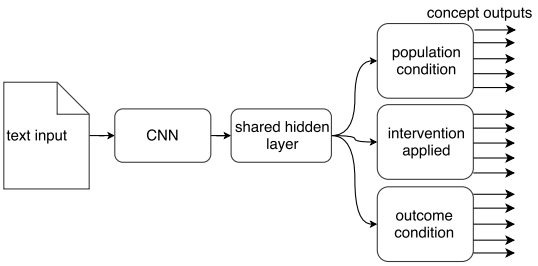


Figure 4.4: The multitask neural architecture we use to directly predict structured vocabulary terms from free texts.

neural *multitask* [99] architecture, depicted in Figure 4.4.[4] In brief, we run input text through a CNN to induce a vector representation, as described in the preceding section. This learned representation is shared across the classification tasks corresponding to the respective PICO elements, thus affording transfer learning across tasks, insofar as the model learns parameters that induce a representation useful for recognizing terms descriptive of the respective PICO elements. Output layers, however, are treated as conditionally independent, given the shared input representation. Thus, e.g., the output layer corresponding to population comprises $|\mathscr{V}|$ binary output nodes (with associated weight vectors) corresponding to concepts in the vocabulary $\mathscr{V}$. Here, $|\mathscr{V}|$ = 366,772. This was prohibitively large, and so as a practical matter we restricted the output size to 150,000 terms (the same 150,000 for each PICO element). These terms include (1) all that appear in the available training sample for any given run, augmented with, (2) terms randomly (IID) sampled from the vocabulary.

4.2.3 Candidate set sampling details

We use the above two methods to generate candidates at test time. Here we describe the training and testing processes related to candidate sampling in greater detail.

During **training**, we draw positive triplets using the ground truth annotations. For example, if we have a set of ground truth annotations $\mathscr{C}_{\mathrm{P}}$, $\mathscr{C}_{\mathrm{I/C}}$ and $\mathscr{C}_{\mathrm{O}}$ for an instance $\mathbf{x}$ then we construct positive triplets $(c_{\mathrm{P}}, c_{\mathrm{I/C}}, c_{\mathrm{O}})$ by randomly and independently sampling one concept each from $\mathscr{C}_{\mathrm{P}}$, $\mathscr{C}_{\mathrm{I/C}}$ and $\mathscr{C}_{\mathrm{O}}$.

We also need to construct negative examples to be fed to the model during

[4] This is similar to the multitask model used in [100].

training. For this we use a 'negative sampling' approach in which we draw one or two concepts from the ground truth set, and the remaining concept(s) from the set of all concepts $\mathscr{V}$. We draw five negative triplets for every positive triplet, and pass these as input to the model in Figure 4.2. In addition to constructing triplets using concepts from the standard vocabulary, we assume the presence of a universal concept "_" in all annotations. This induces triplets of the form of $\{(c_{\mathrm{P}}, \_, \_), (c_{\mathrm{P}}, c_{\mathrm{I/C}}, \_)\}$, in addition to fully specified triplets $(c_{\mathrm{P}}, c_{\mathrm{I/C}}, c_{\mathrm{O}})$. Our CS-Joint model is defined directly over triplets in order to learn the joint distribution of concepts contained in different distinct sets. The introduction of underspecified triplets such as $(c_{\mathrm{P}}, \_, \_)$ effectively allows the model to also learn marginal probabilities of concepts for a particular element, given an input text. We later empirically show the benefit of this approach.

During **testing**, we use the models described in the preceding subsections to generate candidate sets. Specifically, for a given input text, we use MetaMap [10] to generate an unstructured list of candidate terms. We should mention that MetaMap is not trained but comes with pre-fed rules for concept extraction. We also use the multitask model described above (trained on the available training data) to make predictions based on the text, thereby inducing a supplementary, structured candidate set of terms, i.e., these are explicitly associated with individual PICO elements. We then exhaustively construct input candidate tuples by placing the MetaMap candidates into arbitrary slots, combining these with candidates assigned to specific PICO elements by the MT model. In this way, we construct every possible triplet $(c_{\mathrm{P}}, c_{\mathrm{I/C}}, c_{\mathrm{O}})$ that can be derived from the candidate sets; this includes all possible incomplete specifications of the form $(\_, c_{\mathrm{I/C}}, \_)$.

4.3 Experimental Setup

We begin this section by providing details regarding the dataset used for experiments. We then describe the baseline models to which we compare our proposed approaches. Finally, we outline the evaluation setup we adopt and the metrics we use to assess performance.

samples (clinical trials)	4306
distinct population concepts	875
distinct intervention concepts	1115
distinct outcome concepts	1731
population concepts	9387
intervention concepts	5458
outcome concepts	13800

Table 4.1: Dataset statistics.

4.3.1 Dataset

We use a real-world dataset provided by the Cochrane Collaboration[5], which comprises manual annotations applied to biomedical publications. Specifically, aligning with the task we have outlined throughout this chapter, trained annotators have applied tags from a subset of the Unified Medical Language System (UMLS) to free text summaries of biomedical articles, corresponding to the PICO elements. Recall that PICO stands for Population, Intervention/Comparator and Outcomes. These are defined briefly as follows. Population concerns the characteristics of or clinical problem shared by trial participants (e.g., diabetic males). Interventions are the active treatments being studied (e.g., aspirin); Comparators are baseline or alternative treatments to which these are compared (e.g., placebo) – the distinction is arbitrary, and hence we collapse I and C. The outcomes are the variables measured to assess the efficacy of different treatments (e.g., headache severity).

Trained annotators from Cochrane attach concept (UMLS) terms for each PICO element to individual free-text summaries of articles. These summaries comprise fields pertaining to each PICO element for every study. For this work, we merge them into single texts that span all PICO elements; this represents a more typical setup. All collected annotations undergo a rigorous quality assurance process; every annotation is subsequently checked by a domain expert.

4.3.2 Baselines

Two straightforward ways of performing the task under consideration are: (1) simply use MetaMap output, and, (2) train a model that learns to predict UMLS terms for

[5]Cochrane is an international organization that focusses on improving healthcare decisions through evidence: `http://www.cochrane.org/`.

each PICO element directly from the input text.

MetaMap. In the case of using MetaMap, it is not clear how best to assign the unstructured list of terms it provides for a piece of text to the respective PICO elements. Therefore, to make this baseline as competitive as possible, we 'cheat' in its favor by using text explicitly corresponding to different PICO elements. In particular, recall from above that in addition to attaching terms to abstracts, annotators also highlight the text corresponding to each PICO element. Therefore, we know which subspans correspond, e.g., to the population description in a given text. To induce P terms using MetaMap, we then pass *only* this population-specific text to MetaMap and retrieve the corresponding terms that it provides. We emphasize that *only* this baseline model has access to the span-level annotations at test time, which would not generally be available. Therefore, this represents an upper-bound on the performance we can expect to realize using MetaMap alone.

Multitask neural model. As a second baseline, we use the output candidate generation model introduced in Section 4.2.2.2 (and depicted in Figure 4.4). Recall that this is a multitask CNN that directly predicts terms for each PICO element, given the input text.

4.3.3 Evaluation Details

We divided the data into 60/40 for train/test split. We had ground truth annotations for all instances and for all three PICO elements, i.e., all texts have been annotated by domain experts with structured UMLS terms. The texts here are themselves summaries of each element written for previous reviews; we therefore concatenated these together, forming contiguous texts for each instance comprising spans relevant to the respective elements. We used only the 'Cochrane subset' of the UMLS. This is because the annotations we have (performed by Cochrane) contain only terms from this set. The Cochrane vocabulary comprises 366,772 concepts.

All hyper-parameter tuning was performed via nested validation (i.e., within train set). In particular, we used 30% of the training data for hyperparameter tuning. This included iteratively experimenting with and improving the structure of the network. The dropout rate [103] was tuned over a range of 10 equidistant values

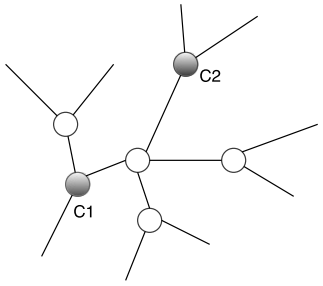


Figure 4.5: We consider two nodes at a distance of less than r hops as an 'r-hop match'; with this we compute the precision@r-hops and recall@r-hops metrics.

in the interval $[0, 1]$. The threshold for binary classification for each term (i.e., the threshold above which a term will be assigned) was tuned over the same range and interval. During hyperparameter search we optimized for average F1-score outputs. We trained for 100 epochs, caching and ultimately using the parameters that performed best on a nested validation set.

As mentioned previously, word embeddings were initialized to pre-trained vectors fit by running word2vec over all biomedical abstracts indexed on MEDLINE.

4.3.4 Metrics

We evaluated the performance of our approach using three standard metrics: precision, recall, and their harmonic mean (i.e., F1 score). We calculated these metrics for each instance and category (i.e., for each PICO element) separately, and aggregated over all instances for the respective categories to obtain MicroPrecision, MicroRecall and MicroF1 scores.

These metrics are strict because they require exact matches between predicted and true concepts. Results will thus be pessimistic in the sense that the model will be heavily penalized for predicting a concept that is semantically similar to (i.e., nearby in the ontology) — but not an *exact* match to — a target concept. As a simple means of relaxing match criteria, we therefore additionally report precision and recall at '2-hops' distance between annotations. Briefly, this counts a predicted term as a match to a target term if the former can reach the latter by taking two hops or fewer. More generally, we also report precision and recall at k hops for varying values of k in Figure 4.6.

Category	Model	Precision	Recall	F1-score	Pr-2hops	Re-2hops	F1-2hops
Population	MetaMap	0.134	0.280	0.181	0.262	0.489	0.341
	Multitask	0.358	0.383	0.370	0.501	0.502	0.501
	CS-Ind	**0.385**	0.529	0.446	**0.557**	0.636	**0.594**
	CS-Cond	0.384	0.535	**0.447**	0.553	0.640	0.593
	CS-Joint	0.318	**0.594**	0.415	0.485	**0.709**	0.576
Interventions/Comparator	MetaMap	0.108	0.288	0.157	0.163	0.387	0.230
	Multitask	0.248	0.245	0.246	0.264	0.262	0.263
	CS-Ind	0.226	0.272	0.247	0.274	0.322	0.296
	CS-Cond	0.225	0.282	0.250	0.275	0.331	0.300
	CS-Joint	**0.265**	**0.421**	**0.326**	**0.314**	**0.473**	**0.378**
Outcomes	MetaMap	0.209	0.391	0.273	0.314	0.518	0.391
	Multitask	0.198	0.211	0.204	0.283	0.290	0.286
	CS-Ind	0.272	0.497	0.352	0.380	0.593	0.464
	CS-Cond	0.268	0.497	0.348	0.378	0.591	0.461
	CS-Joint	**0.279**	**0.503**	**0.359**	**0.38**	**0.595**	**0.468**

Table 4.2: Precisions, recalls and f1 measures realized by different models on the respective PICO elements. Best result for each element and metric are **bolded**. Models with prefix 'CS' (below the dotted lines) are variants of the Candidate-Selector approach we have proposed in this work. We should mention that r-hop refers to the case when we consider a match between two concepts that are at a distance of $\leq r$ hops.

Category	Model	Precision	Recall	F1-score
Population	MetaMap	0.190	0.274	0.224
	Multitask	0.355	0.562	0.435
	CS-Ind	0.413	0.758	0.534
	CS-Cond	**0.490**	0.731	**0.587**
	CS-Joint	0.413	**0.772**	0.539
Interventions	MetaMap	0.119	0.296	0.170
	Multitask	**0.298**	0.371	**0.331**
	CS-Ind	0.162	0.230	0.191
	CS-Cond	0.196	0.250	0.219
	CS-Joint	0.234	**0.420**	0.300
Outcomes	MetaMap	0.270	0.397	0.321
	Multitask	0.339	0.319	0.328
	CS-Ind	0.352	0.560	0.432
	CS-Cond	**0.356**	0.601	0.447
	CS-Joint	0.355	**0.633**	**0.455**

Table 4.3: Results on completely held out data (reference annotations were collected during model development).

Category	Model	Precision	Recall	F1-score	Pr-2hops	Re-2hops	F1-2hops
Population	CS-Joint random	**0.268**	**0.251**	**0.259**	0.386	0.382	0.384
	CS-Joint pre-trained	0.264	0.250	0.257	**0.392**	**0.392**	**0.392**
Interventions/Comparator	CS-Joint random	0.219	0.248	0.233	0.272	**0.294**	**0.283**
	CS-Joint pre-trained	**0.233**	**0.257**	**0.244**	**0.273**	0.293	0.282
Outcomes	CS-Joint random	0.315	0.302	0.308	0.412	0.404	0.408
	CS-Joint pre-trained	**0.341**	**0.356**	**0.348**	**0.440**	**0.449**	**0.445**

Table 4.4: The performance of the CS-Joint model when using randomly initialized versus pre-trained embeddings. Recall from above that the pre-trained embeddings for words were learned using *word2vec* [1] on MEDLINE abstracts, while the concept embeddings were learned using *DeepWalk* [2] over the medical concept vocabulary graph.

Category	Model	Precision	Recall	F1-score	Pr-2hops	Re-2hops	F1-2hops
Population	CS-Joint Complete	0.197	0.145	0.167	0.267	0.216	0.239
	CS-Joint +Marginals	**0.264**	**0.250**	**0.257**	**0.392**	**0.392**	**0.392**
Interventions/Comparator	CS-Joint Complete	0.156	0.149	0.153	0.180	0.168	0.174
	CS-Joint +Marginals	**0.233**	**0.257**	**0.244**	**0.273**	**0.293**	**0.282**
Outcomes	CS-Joint Complete	0.182	0.138	0.157	0.224	0.182	0.201
	CS-Joint +Marginals	**0.341**	**0.356**	**0.348**	**0.440**	**0.449**	**0.445**

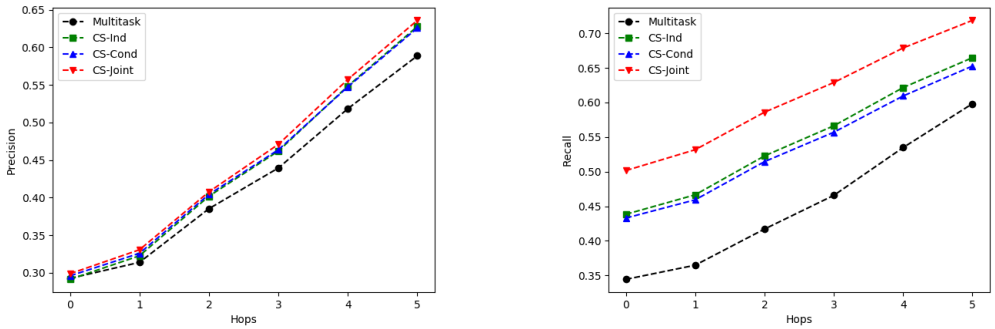
Table 4.5: The performance of the CS-Joint model trained using only completely specified candidate triplets of the form $(c_{\mathrm{P}}, c_{\mathrm{I/C}}, c_{\mathrm{O}})$ (referred to as CS-Joint Complete) versus a variant that accepts partially specified frames like $(\_, \_, c_{\mathrm{O}})$ or $(c_{\mathrm{P}}, \_, c_{\mathrm{O}})$ and marginalizes over missing elements; we refer to the latter approach as CS-Joint +Marginals. As can be seen, the marginals approach yields consistently better predictive results, which is intuitive because it is less restricted, but still exploits correlations. This is the CS-Joint variant that we use.

4.4 Results

4.4.1 Quantitative Results

We report results for all models in Table 4.2. When reading the results here, which are low in absolute terms, it is important to keep in mind two key points. First, the output space is vast, which makes the task inherently quite difficult. And second, as mentioned above, the metrics are pessimistic here because they are very strict in requiring exact (or near-exact, in the case of the 2-hop metrics) matches.

The methods prefixed with 'CS-' (below the dotted lines) are the three instantiations of the Candidate-Selector framework we introduced in Section 4.2; these are compared to the two baselines described in Section 4.3.2. A few observations: CS-approaches uniformly beat baseline strategies, and the gains are considerable: we realize a 7-15 point absolute boost in F1-score, compared to the multitask neural



(a) Precision **(b)** Recall

Figure 4.6: Average (over PICO elements) r-precisions (a) and recalls (b) for each method as a function of r (i.e., using increasingly relaxed metrics; r-precision) counts a predicted concept as matching the truth concept when it is $\leq r$ hops away.

model baseline. We also observe that the CS-Joint approach (Figure 4.2) yields the best performance for both precision and recall (and so also F1) for interventions and outcomes categories, and remains competitive with respect to population predictions (achieving the best recall at a modest cost in precision). This demonstrates the advantage of exploiting correlations between the PICO elements.

Figure 4.6 shows mean r-precisions and r-recalls (mean taken over the three PICO elements) achieved, as a function of r. Thus these plots show the results achieved under increasingly relaxed definitions of concept matches. Note that we omit the MetaMap baseline from these plots because it performed very poorly, to the extent that it rendered the plots difficult to read. The salient observation here is that the CS- models dominate the multitask CNN baseline, and the CS-joint model is consistently the best performing. In other words, the results just reported are robust to more relaxed definitions of concept matches.

4.4.1.1 Unseen Concepts

As mentioned at the outset of this thesis, a challenge in healthcare applications of machine learning is limited training data. In our case, this is compounded by the very large output (label) space. As a consequence, the test data often contains concepts (i.e., labels) that were never seen in the training data. Approaches that learn to directly map from texts to predicted concepts would be generally incapable of

Category	Unseen concepts	Correctly classified
Population	193	24
Intervention	326	54
Outcome	423	77

Table 4.6: The number of unseen concepts identified correctly by the proposed CS-Joint model. The proposed model can identify such unseen concepts due to the use of MetaMap to generate candidate concepts, which may be novel from the perspective of the model. However, our use of pre-trained concept embeddings means that even when previously unseen, the model is sometimes able to correctly select such concepts. Models that explicitly learn to map input texts to concepts will in general be incapable of recognizing concepts not present in the training data. We should mention that the unseen concepts identified by the multi-task classifier (i.e. without MetaMap) were ≈ 0.

predicting unseen concepts, by construction. Thus, e.g., our multitask CNN cannot predict a concept it has never seen in the training data, as there is no means of training the weights parameterizing the node corresponding to the unseen concept. However, because our Candidate-Selector architecture takes as inputs (embeddings of) candidate concepts, these can indeed be completely novel from the models perspective. Our use of MetaMap – and external candidate generator, effectively – means that it is entirely possible to select previously unseen terms. We show this in Table 4.6.

4.4.1.2 Pre-trained vs. Randomly Initialized Concept Embeddings

Recall that we use pre-trained distributed representations of medical concepts, induced via *DeepWalk* [2] performed over the UMLS graph. Here we explore the benefit (if any) of initializing embeddings to pre-trained vectors, as compared to randomly initializing them. In Table 4.4 we report results using these two initialization strategies. In general, using pretrained embeddings for initialization perhaps provides a slight edge, but the differences are not consistent.

4.4.1.3 Marginal vs. 'Complete' CS-Joint variant

Recall (Section 4.2) that the proposed CS-Joint model accepts as input triplets of candidate concepts, each assigned to a particular PICO element. This allows the model to exploit correlations between, e.g., populations and corresponding

interventions. However, we would like to also enable the model to consider marginal probabilities of individual terms, conditioned on the input text). The model should be able to select these when appropriate, regardless of the other PICO term designations. To this end, in Section 4.2.3 we introduced the trick of including partially specified triplets, e.g., $(c_{\text{P}}, c_{\text{I/C}}, \_)\}$. Such partially specified triplets are also considered at test time during our exhaustive consideration of candidate triplets. The alternative would be to use *only* fully specified PICO triplets. To validate the 'marginals' approach adopted, we therefore compared these two strategies. We report results in Table 4.5. Using the partially specified (marginal) triplets clearly and uniformly improves model performance.

4.4.1.4 Results on final heldout data

Finally, we report results achieved by the final models (trained on the entire dataset explored thus far) on a completely new/heldout set of data, collected while we developed the model. This dataset comprises 88 instances, annotated in total with 76, 87, and 139 unique concepts corresponding to population, intervention/comparator and outcomes, respectively.

Results on this dataset are reported in Table 4.3. Here we report only zero-hop measures for brevity, although results with respect to two-hop metrics are comparable. We can see that the proposed CS- models again generally best baselines, and that on average CS-Joint model performs the best of these, achieving a mean F1 across elements of 0.43, versus 0.42 for CS-Cond and 0.37 for the multitask model.

4.4.2 Qualitative Analysis

In addition to the quantitative results reported above, we performed a modest qualitative analysis. In particular, a selection of the model output on the test set was assessed qualitatively by an author who is clinically trained, and by an external annotation quality expert. We provide an illustrative example of model output in Figure 4.7. Qualitatively, the output was deemed usable for information retrieval purposes, and the majority of fields examined were populated with correct concepts. Missing concepts appeared to be the most common error type (e.g. 'Third Trimester

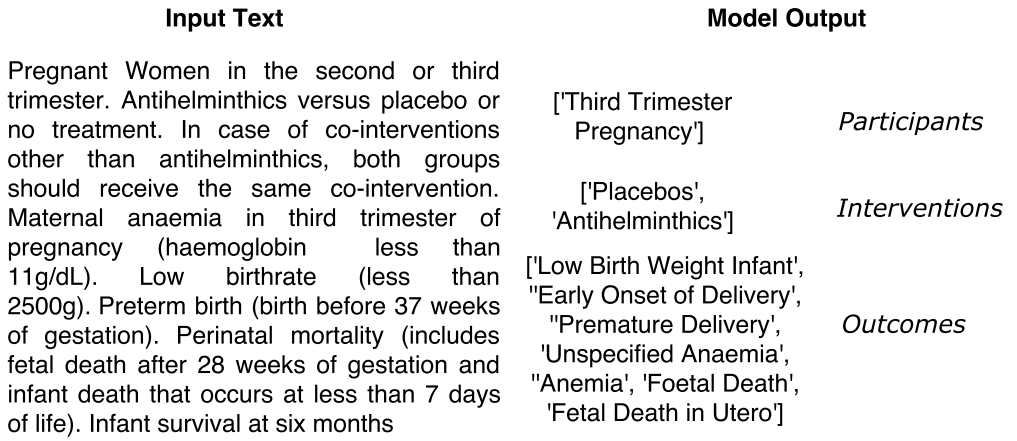


Figure 4.7: Illustrative example of model output.

Pregnancy' was correctly detected in Figure 4.7, but 'Second Trimester Pregnancy' was not); these typically appeared to be caused by a concept not being present in the candidate set generated via MetaMap. Some source texts were short and lacking in detail (particularly those describing outcomes), resulting in missed annotations.

Perhaps unsurprisingly, longer and more descriptive source texts appeared to result in better quality output from MetaMap. Our system currently does not make use of negation information; so, e.g., characteristics of excluded populations would be assigned a positive concept. Overall, the annotations appeared more useful qualitatively than the quantitative results might suggest (given the low absolute values, which we discussed in brief above). We acknowledge that the opinions are subjective to the experts, but the complementary quantitative results are provided in the previous sections.

4.5 Conclusions

We developed a new model for structured clinical text annotation that can work effectively with limited training data. In particular, our model learns to infer terms from the UMLS metathesaurus that describe the individual PICO elements relevant to a given study, as described in an input free-text. This is an important practical task for biomedical natural language processing. Our model defines a novel Candidate-Selector architecture composed of two parts: candidate generation and then (possibly joint) selection and assignment of these candidates to constituent PICO elements. In

our CS-Joint model the selection model is a Convolutional Neural Network jointly conditioned on a triplet of structured PICO UMLS terms and the free-text to be annotated, thus realizing a fully joint approach. This model achieved consistently strong empirical results, beating alternative approaches.

Moving forward, we believe we can further improve upon this model within the same framework, by better exploiting the ontological structure underlying UMLS. We also hope to focus efforts on improving the recognition of novel (unseen) terms, as this is important for the present task.

Chapter 5

Structured Text Tagging via Attentive Neural Tree Decoder

We consider the task of multilabel text annotation, where labels are drawn from an ontology. We should mention that the previous chapter also focused on multilabel annotation but differently in this chapter: 1) we do not apply PICO aspect based tags and 2) the labels are drawn from a strictly tree-structured ontology. As in the previous chapter, we are again motivated by problems in biomedical NLP [82, 81]. Specifically, scientific abstracts in this domain are typically associated with multiple Medical Subject Heading (MeSH) terms. MeSH is a controlled, hierarchically structured vocabulary that facilitates semantic labeling of texts at varying levels of granularity. This in turn supports semantic indexing of biomedical literature, thus facilitating improved search and retrieval.[1].

At present, MeSH annotation is largely performed manually by highly skilled annotators employed by the National Library of Medicine (NLM). Automating this annotation task is thus highly desirable, and there have been considerable efforts to do so. The BIOASQ[2] challenge, in particular, concerns MeSH annotation, and competitive systems have emerged from this in past years [105, 106]; these constitute baseline approaches in the present work.

More generally, MeSH annotation is a specific instance of multi-label classification, which has received substantial attention in general [107, 92, 108, 109,

[1] This problem also resembles tagging clinical notes with ICD codes [104].

[2] `http://bioasq.org/`

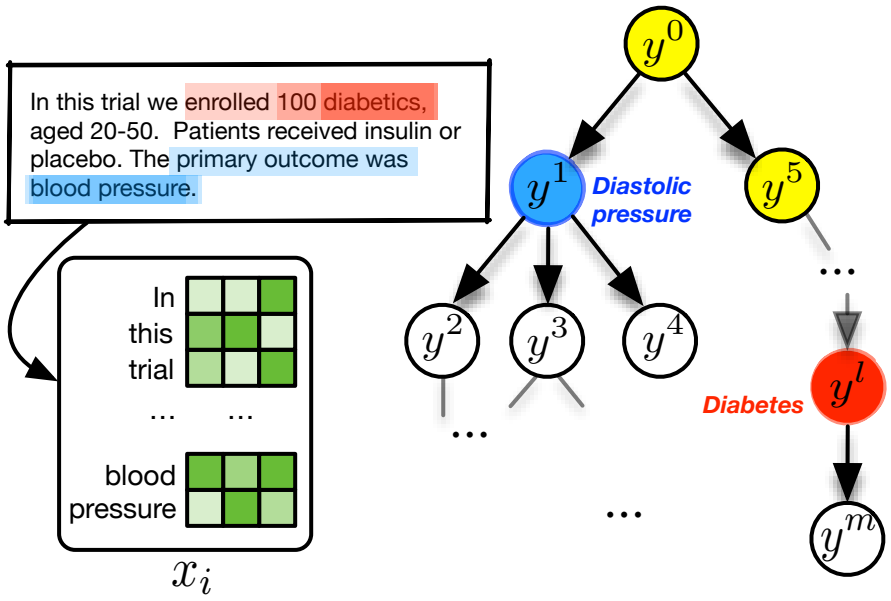


Figure 5.1: Illustration of the proposed Neural Tree Decoding (NTD) model. Input text is encoded, and a decoder then conditionally traverses the label tree to select all relevant nodes to apply, with node-wise attention induced over the input text.

110, 111, 112]. Our work differs from these prior efforts in that MeSH tagging involves *structured* multi-label classification: the label space is a tree[3] in which nodes represent nested semantic concepts, and the specificity of these increases with depth.

Past efforts in multi-label classification have considered hierarchical and tree-based approaches for tagging [112, 113, 110], but these have not assumed a *given* structured label space; instead, these efforts have attempted to *induce* trees to improve inference efficiency. By contrast, we propose to explicitly capitalize on a known output structure codified here by the target ontology from which tags are drawn. We realize this by recursively traversing the tree to make (conditional) binary tag application predictions.

The contribution of this work is a neural sequence-to-sequence (*seq2seq*) model [114] for structured multi-label classification. Our approach entails encoding the input text to be tagged using an RNN, and then *decoding into the ontological output space*. This involves a tree traversal beginning at the root of the tree. At each step,

[3]Technically, MeSH comprises multiple trees, but we join these by insertion of an overarching root node.

the decoder decides whether to 'expand' children as a function of a hidden state vector, node embeddings, and induced attention weights over the input text. This approach is depicted in Figure 5.1. Expanded nodes are added to the predicted tag set. This process is repeated recursively until either leaf nodes are reached or no children are selected for expansion. This neural tree decoding (NTD) model outperforms state-of-the-art models for MeSH tagging.

5.1 Related Works

We are focusing on the problem of multi-label text classification, specifically, when the labels are organized in a tree structured ontology. Consequently, we ought to review two separate threads of relevant previous works: multi-label classification and hierarchical tagging.

5.1.1 Multi-label Classification

We have already provided a literature review of multilabel classification models in the previous chapter, therefore, we are omitting it in this chapter.

5.1.2 Hierarchical Classification

Often times in biomedical domain the label space is structured as hierarchy, for instance, when the labels are drawn from a tree structured vocabulary i.e. an Ontology. In order to leverage the hierarchical structure of labels, there have been attempts at hierarchical classification models [112, 113, 110]. In [112], they propose jointly learning the hierarchical label structure and input representations, as opposed to previous works that learned them in sequence. Similarly, in [113] the authors propose learning a logarithmic depth tree, one that can estimate the conditional probability for a label in $O(\log n)$ time. This reduces a multi-label classification problem into many binary classification problems.

A lot of these methods were focused on learning a tree structure, as opposed to using the one provided beforehand e.g. Ontology. There have been works that diffuse the hierarchical information into the model using label embeddings pre-trained on the provided graph structure. Our work in the previous chapter [115] used label embeddings learned using deepwalk over the ontology graph as inputs to a CNN

based classification network. Another work [51] uses the word-embeddings of labels learned over Wikipedia but for multi-class classification. In [116], they focus on the problem of ICD code assignments to biomedical text e.g. Doctor's notes. These ICD codes encode the revelations in the diagnosis of the patient. These codes are very similar in nature to the MeSH terms. While MeSH terms encode varied biomedical concepts, ICD codes are focused on the diagnostic information e.g. diseases. They propose using a hierarchical bidirectional gated recurrent neural network for tagging discharge summaries of patients with ICD codes. One more work [117] developed a Bayesian framework for combining multiple classifiers based on the functional taxonomy constraints. They used a hierarchy of SVM classifiers trained on multiple data types, they then combined predictions using a Bayesian framework to obtain the most probable set of predictions. We should mention that recursive-tree neural networks [118, 119] have been proposed earlier, but they often focus on recursive structure in the input, as opposed to the output.

5.2 Model

Overview. Our model is an instance of an encoder-decoder architecture. For the encoder, we adopt a standard Gated Recurrent Unit (GRU) network [120], which yields hidden states for the tokens comprising an input document. The decoder network consumes these outputs and begins at the root of the ontological tree. It induces an attention distribution over encoder states, which is used together with the current decoder state vector to inform which (if any) of its immediate children are applicable to the input text (Figure 5.1). This decoding process proceeds recursively for all children deemed relevant. Below we provide more in-depth technical detail regarding the constituent modules.

The encoder (ENC) consumes as input a raw sequence of words, here composing an abstract. These are passed through an embedding layer, producing a sequence of word embeddings x (for clarity we omit a document index here), which are then passed through a GRU [121] to obtain a sequence of hidden vectors $h = \{h_0, \cdots, h_T\}$, where $h_t = \text{GRU}(x_t, h_{t-1})$.

These are then passed to our neural tree decoder, which is responsible for tagging the encoded text with an arbitrary number of terms from the label tree, i.e., sequences in the structured output space. This module traverses the label space top-down, beginning at the root, thus exploiting the concept hierarchy codified by the tree structure.

At each step in the decoding process, the decoder will be positioned at a particular node in the tree n. Children — immediate descendents — of this node are then considered for expansion in turn, based on a hidden state vector s_n, and a context vector c_n. Both of these are initialized to zero vectors and recursively updated during traversal, i.e., as nodes are selected for expansion (and hence added to the predicted tag set). More specifically, the context vector that informs the decision to expand node v in the label hierarchy from its parent node n is a weighted sum of the encoder hidden states h, where weights reflect induced attention over inputs, conditioned on n. That is:

$$c_n = \sum_j \alpha_{nj} h_j \tag{5.1}$$

where

$$\alpha_{nj} = \frac{exp\{a(s_n, h_j)|\theta_n\}}{\sum_l exp\{a(s_n, h_l)|\theta_n\}} \tag{5.2}$$

and a is a simple multi-layer perceptron (MLP), with node-specific parameters θ_n. Here both sums range over the length of the input text.

Given c_n, we then estimate the probability that child label v is applicable to the current input text as a function of the decoder state vector (s_n), where $s_n = \text{GRU}(s_{n-1}, y_n)$, the current context vector (c_n) and the decoder parameters. In particular, this is realized via a standard linear layer with sigmoid activations, parameterized by a weight matrix W comprising independent weight vectors for each output node v. Thus the score for a particular output node v is $\sigma(W_v \cdot [s_n, c_n])$, where W_v denotes the weight vector for output node v.

Pseudocode for the training and decoding procedures are presented in Algorithm

Algorithm 2 RECURSIVETREEDECODING

function NODELOSS(n, h, s, y)
  $l_n \leftarrow 0$
  $c_n, s_n \leftarrow$ DEC(h, n, s)
  for each child $v \in$ children(n) **do**
    $\hat{y}_v \leftarrow \sigma(W_v \cdot [s_n, c_n])$
    $p_v \leftarrow \propto$ depth in tree
    $B_v \sim \text{Ber}(p_v)$
    if B_v **then**
      $l_n \leftarrow l_n + \mathscr{L}(\hat{y}_v, y)$
    end if
    if $\hat{y}_v > \tau$ **then**
      $l_n \leftarrow l_n +$ NODELOSS(v, h, s_n, y)
    end if
  end forreturn l_n
end function

function TRAIN(x, y, α, epochs)
  $\theta \leftarrow$ INIT(θ)
  $e \leftarrow 0$
  while $e <$ epochs **do**
    for each instance $x_i \in x$ **do**
      $h_i \leftarrow$ ENC(x_i)
      $s_0 \leftarrow \mathbf{0}$
      $l_i \leftarrow$ NODELOSS(ROOT, h_i, s_0, y_i)
      $\Delta\theta \leftarrow$ BACKPROP(l_i)
      $\theta \leftarrow \theta + \alpha\Delta\theta$
    end for
    $e \leftarrow e + 1$
  end whilereturn θ
end function

2. In the NODELOSS function, n denotes a particular node. The set of hidden vectors induced by the encoder (corresponding to the inputs) are denoted by h, s is the hidden state of the decoder, and y is the reference label (this encodes a path in the output tree). We assume the decoder, DEC, consumes input representations, a node index and a hidden state and yields a context vector for n, c_n and an updated state vector s_n; in our case the latter is implemented via a GRU. The advantage of using an RNN during decoding is that this allows the exploitation of learned, distributed hidden representations of partial tree paths, which inform node-wise attention and subsequent predictions.

Incurring loss for all nodes along the path specified by y would place a dis-

proportionate amount of emphasis on correctly applying terms that are 'higher' in the ontology, as loss will be propagated for the initial predictions concerning the application of these and then also, due to recursive application, for all of their children (and so on). Thus we only incur (and hence backpropagate) loss for a node v stochastically, according to a Bernoulli distribution B with parameter p_v. We set p_v to be proportional to the depth of node v in the tree such that we are likely to incur larger loss for deeper (rarely occurring) nodes. We operationalize this as: $p_v = min(1, 0.5 + \frac{m}{f_v})$, where m is the count corresponding to the least frequently observed node in the training corpus and f_v is the count for node v. In Section 5.4 we demonstrate the benefit of this approach.

At train time we use *teacher forcing* [122] during decoding. That is, we revert the model back to the correct (training) tree subsequence when it goes off-course, and continue decoding from there. We have elided this detail from the pseudocode for clarity.

5.3 Experimental setup

Below we describe experimental details concerning our implementation, datasets and baselines. Code and data to reproduce our results is available at `https://github.com/gauravsc/NTD`.

5.3.1 Implementation Details

We limited the vocabulary to the $50,000$ most frequent words. Word embeddings were initialized to pre-trained vectors induced via word2vec, trained over a large set of abstracts indexed on PubMed[4]. Ontology node embeddings were pre-trained using DeepWalk [2], fit over PubMed.

5.3.2 Dataset

Our dataset comprises abstracts of articles describing randomized controlled trials (RCTs) from PubMed along with their MeSH terms. The MeSH annotations were manually applied by professionals at the National Library of Medicine (NLM). The

[4] A repository of biomedical literature.

label space underlying MeSH terms is codified by a publicly available ontology.[5]

We split this dataset into disjoint sets for training/development and final evaluation (Table 5.1). We further separated the former into train, validation and development test subsets, to refine our approach. For our final evaluation we used a heldout set of 10,000 abstracts that were not seen in any way during model development and/or hyperparameter tuning. We performed extensive hyperparameter tuning for the baseline models to ensure fair comparison; details regarding this tuning are provided in the Appendix.

5.3.3 Baselines

We compare our proposed approach to three baselines, including two prior winners of the annual BioASQ challenge, which includes an automated MeSH annotation task. However, it is important to note that we used a different (and considerably smaller) dataset in the current work, as compared to the corpus used in the BioASQ challenge. Please also note that all the baselines were trained on the same set of data.

LSSI [106] use an approach that involves predicting both the number of terms and which to apply to a given abstract. They use linear models for both tasks, which operate over TF-IDF representations of abstracts. Specifically, they train a regressor to predict k, the number of MeSH terms to be applied to an abstract. Simultaneously, a binary linear SVM is trained independently for each MeSH term appearing in the train set. At test time, these SVMs provide scores for each term and the top $\hat{k}$ terms are applied, where $\hat{k}$ is the estimate from the aforementioned regressor.

UIUC [105] uses a learning-to-rank model to identify the top MeSH terms for an abstract from a candidate set of terms, which is obtained from the nearest neighbours of the abstract. Additionally, one SVM classifier is trained for each of the MeSH terms (similar to the above approach), and scores for each are used to obtain additional terms to be added to the candidate set. In the end, a threshold (tuned on the validation set) is used to select the final set of terms to be assigned.

Finally, we consider a deep multilabel classification model **DML** [123] that takes

[5] `https://meshb-prev.nlm.nih.gov/treeView`

Train	20000
Validation	4000
Dev test	18884
Test (held-out)	10000
Mean MeSH terms per article	15.33
Total unique MeSH terms	27892
Unique MeSH terms in dataset	3781

Table 5.1: Dataset statistics.

as input unstructured abstracts and activates the output nodes corresponding to the relevant MeSH terms. In brief, embedded tokens are fed through a CNN to induce a vector representation, which is then passed on to the dense output layer. Finally, this is passed through a sigmoid activation function. Note that this model exploits the same pre-trained word embeddings as our model does.

5.3.4 Evaluation metrics

We first evaluate model performance via output node-wise precision, recall and F1 measure. However, these metrics are overly strict in the sense that a model will be penalized equally for all mistakes, regardless of whether they are nearby or far from the target in the label tree. This is problematic because whether to apply a specific MeSH term or its immediate parent may be somewhat subjective in practice [124, 125]. To quantify this, and to explore the extent to which explicitly decoding into the target label space yields improved predictions, we also consider a measure that we refer to as *semantic distance* (SD):

$$\text{SD} = \frac{1}{|\mathscr{Y}|} \sum_{u \in \mathscr{Y}} \min_{v \in \hat{\mathscr{Y}}} dist(u, v) \tag{5.3}$$

where $\mathscr{Y}$ and $\hat{\mathscr{Y}}$ are the sets of target and predicted terms respectively, and *dist* is a function that returns the shortest distance between two nodes in the label ontology tree. The idea is that this penalizes less for 'near misses'. Thus if a model fails to apply a particular tag t, but does apply one near to t in the label tree, then it is penalized less.[6] We hypothesize that our model will improve results markedly with

[6]This metric is equivalent to the sum of two metrics ("divergent path to gold standard" and "divergent path to prediction") defined in [126].

Method	Precision	Recall	F1	SD
LSSI	0.326	0.293	0.309	1.518
UIUC	0.236	**0.388**	0.291	1.433
DML	0.378	0.223	0.275	1.516
NTD-d	**0.434**	0.235	0.299	1.209
NTD-s	0.425	0.265	**0.327**	**1.130**

Table 5.2: Results on the held-out test dataset. SD refers to *semantic distance*, defined in Eq. 5.3.

respect to this metric, given our exploitation of the tree structure.

As in the case of recall, SD can be 'gamed': one can achieve a perfect score by predicting that all nodes apply to a given abstract. Thus this is only meaningful alongside complementary metrics like F1.

5.4 Results

Results on the test set (which was completely held out during development) are reported in Table 5.2. Please note that all the baselines were also trained on the same dataset as our approach. The proposed Neural Tree Decoding model with stochastic backpropagation (NTD-s) beats the most competitive baseline (LSSI) in F1 score by over 2 points.

To explore the effect of backpropagating loss from nodes in proportion to their depth in the ontology, we also include results for a deterministic variant that does not do this, NTD-d. This version does not perform as well, demonstrating the utility of the proposed training approach.

The metrics reported thus far do not account for the structure in the output space. We thus additionally report results with respect to the the semantic distance (SD) metric (Eq. 5.3). We observe a marked performance increase of $\sim$21% over the best performing baseline. This is intuitive given that we are explicitly decoding into the label tree structure, and demonstrates the ability of our model to learn the ontological structure, thereby predicting semantically appropriate terms.

5.5 Conclusions, Discussion & Limitations

We developed a neural attentive sequence tree decoding model for structured multi-label classification where labels are drawn from a known ontology. The proposed method can decode an input text into a tree of labels, effectively using the structure in the output space. We demonstrated that this model outperformed SOTA approaches for the important task of tagging biomedical abstracts with Medical Subject Heading (MeSH) terms on a modestly sized training corpus. Code and data to reproduce these results are available at `https://github.com/gauravsc/NTD`.

One limitation of our model is that it is comparatively slow, due to having to traverse the tree structure during decoding. Prediction speed may not be a major issue in practice, as articles on PubMed could be batch tagged nightly as they arrive. However, slow decoding also means lengthy training (see Appendix, section A.2 for details). For this reason we have here used a modest training set of $\sim$20k abstracts, which is smaller than corpora used in prior work on this task. Given the relative expressiveness of our model, we expect it to benefit substantially from additional training data, moreso than the simpler baseline architectures. But at present this is only a conjecture.

In future work we thus hope to apply this model to larger datasets, and to address the efficiency issue. Concerning the latter, sibling subtrees may be traversed in parallel, conditioned on the hidden state of their parent. Another promising direction would be to move to convolutional encoder and decoder architectures, designing the latter in a way that similarly capitalizes on the label space tree structure.

Chapter 6

Relation Extraction using Explicit Context Conditioning

There are wide applications for Information Extraction in general and Relation Extraction (RE) in particular, which is one reason why relation extraction continues to be an active area of research. Traditionally, a standard RE model would start with entity recognition and then pass the extracted entities as inputs to a separate relation extraction model, which meant that the errors in entity recognition were propagated to RE. This problem was addressed by end-to-end models [127, 128, 129] that jointly learn both Named Entity Recognition (NER) and Relation Extraction (RE).

Generally, these models consist of an encoder followed by a relationship classification (RC) unit [3, 130, 131]. The encoder provides context-aware vector representations for both target entities, which are then merged or concatenated before being passed to the relation classification unit, where a two layered neural network or multi-layered perceptron classifies the pair into different relation types.

Such RE models rely on the encoder to learn 'perfect' context-aware entity representations that can capture complex dependencies in the text. This works well for intra-sentence relation extraction i.e. the task of extracting relation from entities contained in a sentence. As these entities are closer together, the encoder can more easily establish connection based on the language used in the sentence [130, 131]. Additionally, these intra-sentence RE models can use linguistic/syntactical features for an improved performance e.g. shortest dependency path.

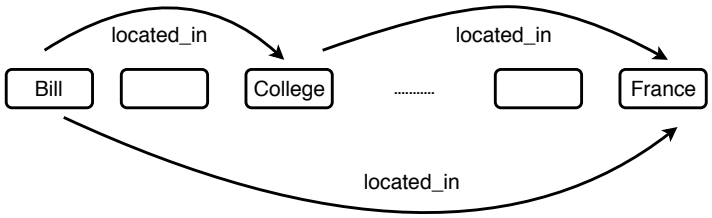


Figure 6.1: Pictorial representation of a second-order relation between two entities (*Bill & France*) connected by a context token (*College*).

Unfortunately, success in intra-sentence RE has not been replicated for cross-sentence RE. As an example, a recent RE method called BRAN [3] proposed to use encoder of Transformer for obtaining token representations and then used these representations for RE. However, our analysis (briefly discussed in Section 6.4.4) revealed that it wrongly marks many cross-sentence relations as negative, especially when the two target entities were connected by a string of logic spanning over multiple sentences. This showed that reliance on the encoder alone to learn these complex dependencies does not work well.

In this work we address this issue of directly using representation generated by the encoder. We propose a model based on the hypothesis that two target entities, whether intra-sentence or cross-sentence, could also be explicitly connected via a third context token (Figure 6.1). More specifically, we find a token in the text that is most related to both target entities, and compute the score for relation between the two target entities as the summation of their relation scores with this token. We refer to these relations as second-order relations. At the end, we combine these second-order scores with first-order scores derived from a traditional RE model, and achieve state-of-the-art performance over two biomedical datasets. To summarize the contribution of this work:

1. We propose using second-order relation scores for improved relation extraction.

2. We describe an efficient algorithm to obtain second-order relation scores.

6.1 Related Works

Relation extraction (RE) has been an active area of research for a long time [132, 133, 134]. and has a rich history of literature. RE models can be broadly classified in two categories based on whether they focus on intra-sentence relations or cross-sentence relations. We will begin with a brief review of past works in relation extraction, going from models that had separate entity recognition to those that jointly extract entities and relations. Afterwards, we will describe in detail two threads of works focused on intra-sentence and cross-sentence RE.

6.1.1 History of Relation Extraction

Prior to end-to-end RE, one would extract the relevant entities, these extracted entities were then given as input to the relation extraction model. The relation extraction algorithm often relied on the information contained in the syntactical structure of sentences. As these grammars are hierarchical, these methods began by constructing dependency parse trees for the sentences from which relations were to be extracted [135, 130, 131, 136]. These trees were then used as inputs for various algorithms. In Zelenko *et al.* [137] authors proposed a kernel based method for extracting relations. It defines a kernel over shallow parse tree representations of unstructured sentences that can compute similarities between two such trees. Another such work [136] used shortest dependency path (SDP) between two words in the dependency tree to establish relations between them. There were many subsequent works that relied on shortest dependency path to extract relations between two words [135, 130, 131].

More recent RE models are based on jointly extracting both entities and relations between those entities i.e. End-to-end RE [127, 128, 129, 138]. These models consist of an encoder followed by a relationship classification (RC) unit [3, 130, 131]. The encoder provides context-aware vector representations for both target entities, which are then merged or concatenated before being passed to the relation classification unit, where a two layered neural network or multi-layered perceptron classifies the pair into different relation types. At the same time the token representations are passed to a separate entity recognition unit.

Another major bottleneck in RE has been the lack of labelled data. To address

this issue there have been attempts at constructing labelled datasets using automated heuristics [139]. The task of learning models using these noisy machine generated datasets is referred to as 'distant supervision'. There have been some more RE models relying on distant supervision that have achieved some degree of success [140, 141].

6.1.2 Intra-Sentence RE

Intra-sentence RE refers to the task of extracting relations between entities are contained within a sentence. There have been a lot of works focused on this problem [130, 131, 142]. Since the entities are contained within a sentence, these intra-sentence RE models can use syntactic or linguistic features for improved RE. An old work in relation extraction [135] proposed a relation extraction model based on using shortest dependency path between target entities in the parse tree. In a recent work [130], the authors propose a random walk based model that learns relation embedding between target entities using an intra-sentence entity graph. Another recent work by Miwa *et al.* [127] proposed an end-to-end RE model that stacks a tree-structured dependency layer on top of a RNN. The dependency layer consists of a tree-LSTM that takes the parse tree as an input. The combination of sequence layer and dependency layer computes dependency-aware and context-aware representations for entities that can then be used for RE. Another work [142] proposed using global co-occurrence statistics of entities for improved RE. These global statistics are (loosely) based on counting the number of times two entities appear in each others' contexts.

6.1.3 Cross-Sentence RE

Cross-sentence RE refers to the task of extracting relations between entities that are not necessarily contained within the same sentence. This is a harder problem in comparison to intra-sentence RE, and therefore has not achieved as much success. A recent cross-sentence RE model [3] proposed to use encoder of Transformer [35] for getting context-aware representations for all the tokens. These representations were then passed to a bilinear relation classification unit. Another work [143] proposed

constructing a document graph that was then passed as an input to a graph-LSTM. This graph-LSTM generated representations for tokens that were then passed to a 2-layered neural network for relation classification. A recent work on cross-sentence relation extraction [144] proposed using distant supervision for RE in cross-sentence scenarios.

6.2 Background

In this section we describe the encoder and relation classification unit of a SOTA RE model called BRAN [3]. This model computes relation scores between two entities directly from their representations, therefore we refer to these as first-order relation scores.

6.2.1 Transformer Encoder

BRAN uses a variant of *Transformer* [35] encoder to generate token representations.

The encoder contains repeating blocks and each such block consists of two sublayers: multi-head self-attention layer followed by position-wise convolutional feedforward layer. There are residual connections and layer normalization [145] after each sublayer. The only difference from a standard transformer-encoder is the presence of a convolution layer of kernel width 5 between two consecutive convolution layers of kernels width 1 in the feedforward sublayer. It takes as input word embeddings that are added with positional embeddings [146].

6.2.2 First-Order Relations

The relation classification unit takes as input token representations from the described encoder. These are then passed through two MLPs to generate head/tail representation e_i^{head}/e_i^{tail} for each token corresponding to whether it serves the first (head) or second (tail) position in the relation.

$$e_i^{head} = W_{head_2}(ReLU(W_{head_1} b_i)) \tag{6.1}$$

$$e_i^{tail} = W_{tail_2}(ReLU(W_{tail_1} b_i)) \tag{6.2}$$

where b_i is the representation of the i_{th} token generated by the encoder.

These are then combined with a bi-affine transformation operator to compute a $N \times R \times N$ tensor A of pairwise affinity scores for every token pair and all relation types, scoring all triplets of the form $(head, relation, tail)$:

$$A_{irj} = (e_i^{head} L) e_j^{tail}, \tag{6.3}$$

where L is a learned tensor of dimension $d \times R \times d$ to map pairs of tokens to scores over each of the R relation types and d is the dimension of head/tail representations. Going forward we will drop the subscript r for clarity.

The contributions from different mention pairs are then aggregated to give us **first-order** relation scores. This aggregation is done using *LogSumExp*, which is a smooth approximation of *max* that prevents sparse gradients:

$$\text{scores}^{(1)}(p^{head}, p^{tail}) = \log \sum_{\substack{i \in P^{head} \\ j \in P^{tail}}} \exp(A_{ij}), \tag{6.4}$$

where $P^{head}(P^{tail})$ contains mention indices for head (tail) entity.

6.3 Proposed Second-Order Relations

In this section we describe in detail our proposed method to obtain second-order relation scores.

We use the encoder described in Sec 6.2.1 for getting token representations. These token representations are then passed through two MLPs (as in previous section), which generate head/tail representations for each token corresponding to whether it serves the first or the second position in the relation. We used a separate set of these head/tail MLPs for second-order scores than the ones used for getting first-order scores. This was motivated by the need for representations focused on establishing relations with context tokens, as opposed to first-order relations (described in previous section) that attempt to directly connect two target entities.

The head and tail representations are then combined with a $d \times R \times d$ bilinear

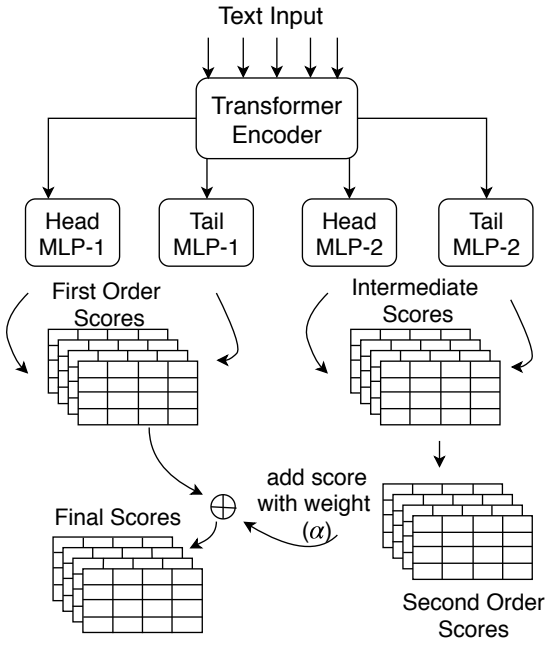


Figure 6.2: Schematic of the model architecture.

transformation tensor M to get a $N \times R \times N$ tensor B of intermediate pairwise scores.

$$B_{ij} = (e_i^{head} M) e_j^{tail} \tag{6.5}$$

After that we arbitrarily define the scores between tokens i and j when *conditioned* on a context token k as the sum of the scores of relations (i,k) and (k,j).

$$C(i,j|k) = B_{ik} + B_{kj} \tag{6.6}$$

These context-conditioned scores are computed for every triplet of the form (i,j,k).

Second-order relation scores are then derived by aggregating over all context tokens and mention pairs using *LogSumExp*.

$$\text{scores}^{(2)}(p^{head}, p^{tail}) = \log \sum_{\substack{k \\ i \in P^{head} \\ j \in P^{tail}}} \exp(C(i,j|k)) \tag{6.7}$$

Here *LogSumExp* ensures that one specific mention pair connected via one or few specific context token(s) is responsible for the relation. This is used to aggregate

weak signals from different context tokens that could potentially connect the two target entities, which reduces over-fitting by reducing contributions from noisy associations of the target entities with random tokens e.g. stopwords.

It is important to mention that a naive implementation of this would require $O(N^3)$ space to store context-conditioned scores between all pairs of token i.e. $C(i, j|k)$. To address this, we describe an efficient method in Section 6.3.1 that avoids explicitly storing these.

At the end, the final score for relation between two entities is given as a weighted sum of first (eq. 6.4) and second (eq. 6.7) order scores.

$$\text{scores}(p^{head}, p^{tail}) = \text{scores}^{(1)}(p^{head}, p^{tail}) + \alpha * \text{scores}^{(2)}(p^{head}, p^{tail}) \quad (6.8)$$

where α is a hyper-parameter denoting the weight of second-order relation scores.

Entity Recognition. We do entity recognition alongside relation extraction, as the transfer of knowledge between the two tasks has been shown to improve relation extraction performance [3, 127]. For this we feed encoder output b_i to a linear classifier W_{er} that predicts scores for each entity type.

$$d_i = W_{er}(b_i) \quad (6.9)$$

6.3.1 Efficient Implementation

The problem lies in storing score for every intermediate relation of the form $C(i, j|k)$, as that would require space of the order $O(N^3)$. Here we describe a space-time efficient method to compute final second-order relation scores.

The intermediate scores (eq. 6.5) are a tensor of dimension $b \times N \times R \times N$ comprising of pairwise scores for b batches. We create two tensors out of these intermediate scores, namely T_1 and T_2. T_1 computes the exponential of indices ($\{b, i \in P^{head}, j \in \mathscr{C}, R\}$) corresponding to pairwise scores between head entity and all the context tokens ($\mathscr{C}$ i.e., all the tokens except the two target entities), and sets other indices to 0. Similarly, T_2 computes exponential of indices ($\{b, i \in P^{tail}, j \in$

Data	Model	Pr	Re	F1
DCN	BRAN	0.614	0.850	0.712
	+ SOR	**0.643**	**0.879**	**0.734**
i2b2	HDLA	0.378	**0.422**	0.388
	BRAN	0.396	0.403	0.395
	+ SOR	**0.424**	0.419	**0.407**
CDR	BRAN	0.552	0.701	0.618
	+ SOR	0.552	0.701	0.618

Table 6.1: The performance of proposed model using second-order relations. BRAN is the model used in [3] and +SOR is our proposed model with second-order relations. Results for HDLA are quoted from [4]. Results on CDR are identical for both BRAN and our proposed model as α was set to 0 after tuning over the dev set at which point our model is the same as BRAN. All the metrics are macro in nature.

$\mathscr{C}, R\}$) corresponding to pairwise scores between tail entity and context tokens, setting all other indices to 0. To get the context conditioned scores one needs to compute the batch product of R two dimensional slices of size $N \times N$ from T_1 and T_2 along the dimension of context, but this would be sequential in R. Instead we can permute T_1 and T_2 to $b \times R \times N \times N$ followed by reshaping to $bR \times N \times N$ and perform a batch matrix multiplication along the context dimension to get $bR \times N \times N$. Afterwards, we can sum along the last two dimensions to get a tensor of size bR. Finally, we can take the log succeeded by reshaping to $b \times R$ to obtain second-order scores.

6.4 Experimentation

6.4.1 Datasets

We have used three datasets in this work, i2b2 2010 challenge [147] dataset, a de-identified clinical notes dataset and a chemical-disease relations dataset known as BioCreative V (CDR) [148, 149].

First is a publicly available subset of the dataset used for the i2b2 2010 challenge. It consists of documents describing relations between different diseases and treatments. Out of the 426 documents available publicly, 10% are used each for both dev and test and the rest for training. There are 3244/409 relations in train/test set and 6 pre-defined relations types including one negative relation e.g. TrCP (**Tr**eatment

Causes **Pr**oblem), TrIP (Tr Improves Pr), TrWP (Tr Worsens Pr). We have used the exact same dataset as Chikka *et al.* [4].

Second is a *Amazon*-owned dataset of 4200 de-identified clinical notes (DCN), with vocabulary size of 50K. It contains approximately 170K relations in the train set and 50K each in dev/test set. There are 7 pre-defined relation types including one negative relation type. These are mostly between medication name and other entities e.g. "*paracetamol* ***every*** *day*","*aspirin* ***with_dosage*** *100mg*". The frequency of different relations in this dataset is fairly balanced.

Third is a widely used and publicly available dataset called CDR [148, 149]. It was derived from Comparative Toxicogenomics Database (CTD) and contains documents describing the effect of chemicals (drugs) on diseases. There are only two relation types between any two target entities i.e. positive/negative and these relations are annotated at the document level. It consists of 1500 documents that are divided equally between train/dev/test sets. There are 1038/1012/1066 positive and 4280/4136/4270 negative relations in train/dev/test sets respectively. We performed the same preprocessing as done in BRAN [3].

6.4.2 Experimental Settings

We jointly solve for NER and RE tasks using cross-entropy loss. During training we alternate between mini-batches derived from each task. We fix the learn rate to 0.0005 and clip gradient for both tasks at 5.0. For training, we used adams optimizer with $\beta = (\beta_1, \beta_2) = (0.1, 0.9)$. We tune over the weight of second-order relations denoted by α to get $\alpha = 0.2$ for DCN/i2b2 and $\alpha = 0.0$ for CDR dataset.

Our final network had two encoder layers, with 8 attention heads in each multi-head attention sublayer and 256 filters for convolution layers in position-wise feedforward sublayer. We used dropout with probability 0.3 after: embedding layer, head/tail MLPs, output of each encoder sublayer. We also used a word dropout with probability 0.15 before the embedding layer.

6.4.3 Results

To show the benefits of using second-order relations we compared our model's performance to BRAN. The two models are different in the weighted addition of

second-order relation scores. We tune over this weight parameter on the dev set and observed an improvement in MacroF1 score from 0.712 to 0.734 over DCN data and from 0.395 to 0.407 over i2b2 data. For further comparison a recently published model called HDLA [4] reported a macro-F1 score of 0.388 on the same i2b2 dataset. It should be mentioned that HDLA used syntactic parsers for feature extraction but we do not use any such external tools.

In the case of CDR dataset we obtained $\alpha = 0$ after tuning, which means that the proposed model converged to BRAN and the results were identical for the two models. These results are summarized in Table 6.1.

Also, we observe that the efficient implementation that we described in Section 6.3.1 reduces the run-time by approximately 3-times when compared to the naive implementation.

6.4.4 Ablation Study

We experimented with different ablations of BRAN and noticed an improvement in results for DCN dataset upon removing multi-head self-attention layer. Also, our (manual) qualitative analysis of 20 documents showed that relations between distant entities were often (i.e. in more than 18 documents) wrongly marked negative. We attribute these errors to the token representations generated by the encoder. To this effect, our experiments showed that incorporating relative position [150] information in the encoder to improve token representations does not lead to superior RE. Separately, we observed that the proposed method improved results when using a standard CNN encoder as well.

6.5 Conclusions and Future Work

We proposed a method that uses second-order relation scores to capture long dependencies for improved RE. These relations are derived by explicitly connecting two target entities via a context token. These second-order relations (SORs) are then combined with traditional relation extraction models, leading to state-of-the-art performance over two biomedical datasets. We also describe an efficient implementation for obtaining these SORs.

Despite restricting ourselves to SORs, it should be noted that the proposed method can be generalized to third and fourth order relations. We conjecture that these may serve well for cross-sentence relation extraction in long pieces of texts. Also, we only considered one relation type between each entity and bridge token but it is possible, and very likely that two different relation types may lead to a third relation type. We will explore both these aspects in future work.

Chapter 7

Conclusions, Future Works & Discussion

7.1 Conclusions

There is a lot of raw text available in the world today. This text is generated everyday in the form of blogs, articles, published papers, social media posts, websites and from millions of other sources. There is a lot of information contained in this data that could potentially be used across various domains with wide applications. The problem is that most of this data is generated by humans, and does not contain any structure i.e. it is raw unstructured text. This raw text can not be used in its current form for most tasks. The challenge is to be able to extract structured information from this raw text. This information extraction task can be performed by human beings but they are expensive. We need to be able to extract this information automatically. This motivates the task of automatic information extraction from text. One way to extract information from text is through *Text Classification*.

In this thesis, we work towards solving various challenges faced during automatic text classification. One such challenge is the problem of zero-shot labels in the text, these are labels that are absent in the training set but can appear in the test set. This happens when the training data is small and the output label space is large, which inevitably leads to many of the labels being missing in the training set. These zero-shot labels are especially an issue during the initial phases of online

web applications when the training data is scarce, or in biomedical domain where obtaining labelled training data is expensive. We deal with this issue of zero-shot labels by proposing a neighbourhood sensitive classification model that relies on fine-tuned semantic label embeddings. These embeddings are fine-tuned on the task specific data, and lead to superior performance. The neighbourhood sensitive model learns to map input text into these fine-tuned embeddings, while taking into account the neighbourhood of the label to reduce probability of misclassification.

Afterwards, we work towards improving active learning for the task of 'citation screening'. There needs to be a survey of relevant literature before formulating policies/guidelines on medical or public-health issues. For this, one needs to identify relevant studies before summarizing their contents. This task of identifying relevant studies for a systematic review is called citation screening. There have been attempts to semi-automate this task using an active learning algorithm. We propose a novel algorithm that explores the topic space during initial phases to be able to exploit the knowledge gained during the final phase of the algorithm. We also perform experiments for choosing the best feature extraction model for active learning in systematic reviews, and describe an effective way to choose the most appropriate feature extraction model for any given review.

After this, we present our work on automatic structured text annotation. Here, we focus on the task of PICO annotations for biomedical paper abstracts. This task poses two unique challenges, one is small training data, and the other is a vast and structured label space. We solve these issues using a neural candidate-selector architecture with help from a biomedical software called 'Metamap' for generating high-recall candidate PICO concepts, which are then filtered, while using correlations among different aspects, to obtain final predictions. In the following work we propose a neural tree decoder for biomedical text tagging, but this time the labels are assumed to be drawn from an Ontology. This model performs better than state-of-the-art approaches for biomedical text tagging by predicting ontologically coherent labels.

At the end, we propose a novel state-of-the-art model for the task of classifying

pairs of words based on the relation between them i.e. Relation Extraction. For this, we propose using second-order relations that are derived by explicitly connecting words via intermediate context tokens, as opposed to only relying on the encoder representations. We also provide an efficient implementation for extracting these indirect relations from the text. This method can also be generalized to extract 3rd and 4th-order relations between entities.

7.2 Future Works

There are still many gaps left in the state-of-the-art that we would focus on in the future. Our neural tree decoder model beats baselines for the task of biomedical text tagging but is comparatively slower. Due to this we were not able to train the model over the entire labeled set of over 10+ million biomedical abstracts that are freely available on Pubmed [1]. We conjectured in our paper that given the complexity of our model we are going to benefit from the larger labeled dataset. We will work towards fixing this major flaw in the future. For this we could parallelize decoding sibling trees to leverage multiple GPUs for speeding up decoding. The other promising direction would be to move towards a *non-recursive* convolutional encoder-decoder architecture. We believe that using a variant of *Transformer* networks for seq-to-tree modelling might lead to significant speed up while still leveraging the Ontology.

Also, we restricted ourselves to second-order relations when extracting relations from unstructured text, but our proposed method can be generalized to third and fourth-order relations. We conjecture that these may serve well for cross-sentence RE in long pieces of texts. Also, we only considered one relation type between each entity and bridge token but it is possible, and very likely that two different relation types may lead to a third relation type. We will explore both these aspects in the future.

7.3 Discussion

A lot of big strides have been made in Natural Language Processing (NLP) in the recent past. Few of the notables ones have been the use of bi-directional feedforward

[1] A repository of biomedical literature

encoders (i.e. Transformer) that are computationally efficient and state-of-the-art in performance. These encoders are orders of magnitude faster than RNNs. Consequently, they have also made it possible to train these networks on larger datasets, which in turn has led to good performance on some benchmark tasks.

However, one major issue with most NLP tasks has been the lack of labelled data required by sophisticated deep models. To address this issue, there have been recent attempts at using transfer learning. The above mentioned bi-directional encoders are pre-trained on extremely large unlabelled datasets e.g. Wikipedia, on subset of language modelling tasks. These pre-trained encoders are appended with task specific layers and the entire network is then fine-tuned on the target task. These fine-tuned networks have achieved remarkable performance over many benchmark task, even surpassing human performance on some of them. Separately, there have also been attempts for automated generation of labelled datasets using heuristic techniques referred to as 'distant supervision'.

Going forward, we think that manually constructing task-specific labelled datasets might become less popular. We can not keep constructing expensive labelled datasets for every task that needs automating. We think the future of text classification, and more generally of NLP, lies with more effective transfer learning. For instance, there exist labelled biomedical datasets e.g. MeSH tagged Pubmed abstracts, that can be used for pre-training deep neural networks followed by fine-tuning on the target tagging tasks e.g. PICO annotation.

Unfortunately, one issue with these pre-trained models is that they require a significant amount of time for fine-tuning. It is because the entire network needs to be fine-tuned on the target task. On the other hand, simply fine-tuning the task specific layers leads to reduced performance. We believe this is because the representations learned by the network are still not general enough owing to the task specific attention network. The self-attention component in these networks is pre-trained on a specific set of tasks and pays attention to those areas of input that correspond to those task(s). We think the network could perhaps be trained with multiple separate attention modules, each corresponding to a different task. At test time, we could choose one

of those attention modules depending on the target task.

A lot of important tasks discussed in this thesis that require automation can benefit from the above mentioned directions of research. By automating these tasks we can save a lot of money for various public institutions that require extracting this information e.g. Governments or Policy makers, which can be of huge benefit to society.

Bibliography

[1] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the Conference on Neural Information Processing Systems*, pages 3111–3119, 2013.

[2] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 701–710, 2014.

[3] Patrick Verga, Emma Strubell, and Andrew McCallum. Simultaneously self-attending to all mentions for full-abstract biological relation extraction. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 872–884, 2018.

[4] Veera Raghavendra Chikka and Kamalakar Karlapalem. A hybrid deep learning approach for medical relation extraction. *KDD Workshop on Machine Learning for Medicine and Healthcare*, 2018.

[5] Ziming Zhang and Venkatesh Saligrama. Zero-shot learning via semantic similarity embedding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4166–4174, 2015.

[6] Mark Palatucci, Dean Pomerleau, Geoffrey E Hinton, and Tom M Mitchell. Zero-shot learning with semantic output codes. In *Proceedings of the Conference on Neural Information Processing Systems*, pages 1410–1418, 2009.

[7] Jingqing Zhang, Piyawat Lertvittayakumjorn, and Yike Guo. Integrating semantic knowledge to tackle zero-shot text classification. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1031–1040, 2019.

[8] Pushpankar Kumar Pushp and Muktabh Mayank Srivastava. Train once, test anywhere: Zero-shot learning for text classification. *arXiv preprint arXiv:1712.05972*, 2017.

[9] Kazuma Hashimoto, Georgios Kontonatsios, Makoto Miwa, and Sophia Ananiadou. Topic detection using paragraph vectors to support active learning in systematic reviews. *Journal of Biomedical Informatics*, 62:59–65, 2016.

[10] Alan R Aronson. Effective mapping of biomedical text to the umls metathesaurus: the metamap program. In *Proceedings of the American Medical Information Association Symposium*, page 17, 2001.

[11] G Zhou, J Su, J Zhang, and M Zhang. Combining various knowledge in relation extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 427–434, 2005.

[12] Zhou GuoDong, Su Jian, Zhang Jie, and Zhang Min. Exploring various knowledge in relation extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 427–434, 2005.

[13] Byron C Wallace, Kevin Small, Carla E Brodley, and Thomas A Trikalinos. Active learning for biomedical citation screening. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 173–182, 2010.

[14] Byron C Wallace, Thomas A Trikalinos, Joseph Lau, Carla Brodley, and Christopher H Schmid. Semi-automated screening of biomedical citations for systematic reviews. *BMC Bioinformatics*, 11(1):1–11, 2010.

[15] Byron C Wallace, Kevin Small, Carla E Brodley, Joseph Lau, and Thomas A Trikalinos. Deploying an interactive machine learning system in an evidence-based practice center: abstrackr. In *Proceedings of the ACM SIGHIT International Health Informatics Symposium*, pages 819–824, 2012.

[16] Makoto Miwa, James Thomas, Alison O'Mara-Eves, and Sophia Ananiadou. Reducing systematic review workload through certainty-based screening. *Journal of Biomedical Informatics*, 51:242–253, 2014.

[17] Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification*. John Wiley & Sons, 2012.

[18] Richard O Duda and Peter E Hart. *Pattern classification and scene analysis*, volume 3. Wiley New York, 1973.

[19] Norman R Draper and Harry Smith. *Applied regression analysis*, volume 326. John Wiley & Sons, 1998.

[20] Anil K Jain and Richard C Dubes. *Algorithms for clustering data*, volume 6. Prentice hall Englewood Cliffs, 1988.

[21] Thomas M Cover and Peter E Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

[22] Byron P Roe, Hai-Jun Yang, Ji Zhu, Yong Liu, Ion Stancu, and Gordon McGregor. Boosted decision trees as an alternative to artificial neural networks for particle identification. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 543(2-3):577–584, 2005.

[23] Baoxun Xu, Xiufeng Guo, Yunming Ye, and Jiefeng Cheng. An improved random forest classifier for text categorization. *Journal of Computers*, 7(12):2913–2920, 2012.

[24] Nello Cristianini and John Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press, 2000.

[25] Peter Bartlett and John Shawe-Taylor. Generalization performance of support vector machines and other pattern classifiers. *Advances in Kernel Methods for Support Vector Learning*, pages 43–54, 1999.

[26] Thorsten Joachims. Training linear svms in linear time. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 217–226, 2006.

[27] David McAllester. Generalization bounds and consistency. *Predicting Structured Data*, pages 247–261, 2007.

[28] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[29] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the Institute of Electrical and Electronics Engineers*, 86(11):2278–2324, 1998.

[30] Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):855–868, 2008.

[31] Li Deng. A tutorial survey of architectures, algorithms, and applications for deep learning. *APSIPA Transactions on Signal and Information Processing*, 3, 2014.

[32] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[33] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[34] Richard Socher, Yoshua Bengio, and Christopher D Manning. Deep learning for nlp (without magic). In *Tutorial Abstracts of Association for Computational Linguistics*, page 5, 2012.

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the Conference on Neural Information Processing Systems*, pages 5998–6008, 2017.

[36] Ioannis Katakis, Grigorios Tsoumakas, and Ioannis Vlahavas. Multilabel text classification for automated tag suggestion. *ECML/PKDD Discovery Challenge*, pages 75–83, 2008.

[37] JDM Rennie. Improving multiclass text classification with the support vector machines. *MIT Artificial Intelligence Laboratory Publications*, 2001.

[38] Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. Zero-shot learning through cross-modal transfer. In *Proceedings of the Conference on Neural Information Processing Systems*, pages 935–943, 2013.

[39] Judit Bar-Ilan, Snunith Shoham, Asher Idan, Yitzchak Miller, and Aviv Shachak. Structured versus unstructured tagging: a case study. *Online Information Review*, 32(5):635–647, 2008.

[40] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the Conference on Neural Information Processing Systems*, pages 3111–3119, 2013.

[41] Bharath Hariharan, SVN Vishwanathan, and Manik Varma. Efficient max-margin multi-label classification with applications to zero-shot learning. *Machine Learning*, 88(1-2):127–155, 2012.

[42] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):453–465, 2014.

[43] Thomas Mensink, Efstratios Gavves, and Cees GM Snoek. Costa: Co-occurrence statistics for zero-shot classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2441–2448, 2014.

[44] Dinesh Jayaraman and Kristen Grauman. Zero-shot recognition with unreliable attributes. In *Proceedings of the Conference on Neural Information Processing Systems*, pages 3464–3472, 2014.

[45] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 951–958, 2009.

[46] Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing objects by their attributes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1778–1785, 2009.

[47] Felix X Yu, Liangliang Cao, Rogério Schmidt Feris, John R Smith, and Shih-Fu Chang. Designing category-level attributes for discriminative visual recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 771–778, 2013.

[48] Marcus Rohrbach, Michael Stark, and Bernt Schiele. Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1641–1648, 2011.

[49] Naman Turakhia and Devi Parikh. Attribute dominance: What pops out? In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1225–1232, 2013.

[50] Devi Parikh and Kristen Grauman. Relative attributes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 503–510, 2011.

[51] Mohammad Norouzi, Tomas Mikolov, Samy Bengio, Yoram Singer, Jonathon Shlens, Andrea Frome, Greg S Corrado, and Jeffrey Dean. Zero-shot learning by convex combination of semantic embeddings. *International Conference on Learning Representations*, 2014.

[52] Bernardino Romera-Paredes and PHS Torr. An embarrassingly simple approach to zero-shot learning. In *Proceedings of the International Conference on Machine Learning*, pages 2152–2161, 2015.

[53] Xin Li, Yuhong Guo, and Dale Schuurmans. Semi-supervised zero-shot classification with label representation learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4211–4219, 2015.

[54] Xin Li and Yuhong Guo. Max-margin zero-shot learning for multi-class classification. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 626–634, 2015.

[55] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543, 2014.

[56] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[57] Tom M Mitchell, Svetlana V Shinkareva, Andrew Carlson, Kai-Min Chang, Vicente L Malave, Robert A Mason, and Marcel Adam Just. Predicting

human brain activity associated with the meanings of nouns. *Science*, 320(5880):1191–1195, 2008.

[58] Arkaitz Zubiaga. Enhancing navigation on wikipedia with social tags. *arXiv preprint arXiv:1202.5469*, 2012.

[59] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Effective and efficient multilabel classification in domains with large number of labels. In *Proceedings of ECML/PKDD 2008 Workshop on Mining Multidimensional Data*, pages 30–44, 2008.

[60] David Gough, Sandy Oliver, and James Thomas. *An introduction to systematic reviews*. Sage, 2012.

[61] Iain Chalmers, Larry V Hedges, and Harris Cooper. A brief history of research synthesis. *Evaluation & the Health Professions*, 25(1):12–37, 2002.

[62] Alison O'Mara-Eves, James Thomas, John McNaught, Makoto Miwa, and Sophia Ananiadou. Using text mining for study identification in systematic reviews: a systematic review of current approaches. *Systematic Reviews*, 4(1):1, 2015.

[63] Aaron M Cohen, William R Hersh, K Peterson, and Po-Yin Yen. Reducing workload in systematic review preparation using automated citation classification. *Journal of the American Medical Informatics Association*, 13(2):206–219, 2006.

[64] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the International Conference on Machine Learning*, pages 1188–1196, 2014.

[65] Yong Cheng Wu. Active learning based on diversity maximization. In *Proceedings of Applied Mechanics and Materials*, pages 2548–2552, 2013.

[66] Klaus Brinker. Incorporating diversity in active learning with support vector machines. In *Proceedings of the International Conference on Machine Learning*, pages 59–66, 2003.

[67] Zuobing Xu, Ram Akella, and Yi Zhang. Incorporating diversity and density in active learning for relevance feedback. In *Proceedings of the European Conference on Information Retrieval*, pages 246–257, 2007.

[68] James Thomas, John McNaught, and Sophia Ananiadou. Applications of text mining within systematic reviews. *Research Synthesis Methods*, 2(1):1–14, 2011.

[69] Chris C Beahler, Jennifer J Sundheim, and Naomi I Trapp. Information retrieval in systematic reviews: challenges in the public health arena. *American Journal of Preventive Medicine*, 18(4):6–10, 2000.

[70] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3(Jan):993–1022, 2003.

[71] Burr Settles. Active learning literature survey. *Science*, 10(3):237–304, 1995.

[72] Manali Sharma, Di Zhuang, and Mustafa Bilgic. Active learning with rationales for text classification. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 441–451, 2015.

[73] Yuanhan Mo, Georgios Kontonatsios, and Sophia Ananiadou. Supporting systematic reviews using lda-based document representations. *Systematic Reviews*, 4(1):1, 2015.

[74] Andrew M Dai, Christopher Olah, and Quoc V Le. Document embedding with paragraph vectors. *arXiv preprint arXiv:1507.07998*, 2015.

[75] Zhengping Che, David Kale, Wenzhe Li, Mohammad Taha Bahadori, and Yan Liu. Deep computational phenotyping. In *Proceedings of the ACM SIGKDD*

International Conference on Knowledge Discovery and Data Mining, pages 507–516, 2015.

[76] Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F Stewart, and Jimeng Sun. Doctor ai: Predicting clinical events via recurrent neural networks. In *Proceedings of Conference on Machine Learning for Healthcare*, pages 301–318, 2016.

[77] Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter Stewart. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. In *Proceedings of the Conference on Neural Information Processing Systems*, pages 3504–3512, 2016.

[78] Zachary C Lipton, David C Kale, Charles Elkan, and Randall Wetzell. Learning to diagnose with lstm recurrent neural networks. *arXiv preprint arXiv:1511.03677*, 2015.

[79] Florian Boudin, Lixin Shi, and Jian-Yun Nie. Improving medical information retrieval with pico element detection. In *Proceedings of the European Conference on Information Retrieval*, pages 50–61, 2010.

[80] Alan R Aronson and François-Michel Lang. An overview of metamap: historical perspective and recent advances. *Journal of the American Medical Informatics Association*, 17(3):229–236, 2010.

[81] D Demner-Fushman, N Elhadad, et al. Aspiring to unintended consequences of natural language processing: A review of recent developments in clinical and consumer-generated text processing. *Yearbook of Medical Informatics*, pages 224–233, 2016.

[82] Pierre Zweigenbaum, Dina Demner-Fushman, Hong Yu, and Kevin B Cohen. Frontiers of biomedical text mining: current progress. *Briefings in Bioinformatics*, 8(5):358–375, 2007.

[83] Svetlana Kiritchenko, Berry de Bruijn, Simona Carini, Joel Martin, and Ida Sim. Exact: automatic extraction of clinical trial characteristics from journal publications. *BMC Medical Informatics and Decision Making*, 10(1):56, 2010.

[84] Rodney L Summerscales, Shlomo Argamon, Shangda Bai, Jordan Hupert, and Alan Schwartz. Automatic summarization of results from clinical trials. In *Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine*, pages 372–377, 2011.

[85] Florian Boudin, Jian-Yun Nie, Joan C Bartlett, Roland Grad, Pierre Pluye, and Martin Dawes. Combining classifiers for robust pico element detection. *BMC medical Informatics and Decision Making*, 10(1):29, 2010.

[86] Byron C Wallace, Joël Kuiper, Aakash Sharma, Mingxi Brian Zhu, and Iain J Marshall. Extracting pico sentences from clinical trial reports using supervised distant supervision. *Journal of Machine Learning Research*, 17(132):1–25, 2016.

[87] Iain J Marshall, Joël Kuiper, and Byron C Wallace. Automating risk of bias assessment for clinical trials. In *Proceedings of the ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 88–95, 2014.

[88] Iain J Marshall, Joël Kuiper, and Byron C Wallace. Robotreviewer: evaluation of a system for automatically assessing bias in clinical trials. *Journal of the American Medical Informatics Association*, 23(1):193–201, 2016.

[89] Ye Zhang, Iain J. Marshall, and Byron C. Wallace. Rationale-augmented convolutional neural networks for text classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 795–804, 2016.

[90] Louise AC Millard, Peter A Flach, and Julian Higgins. Machine learning to assist risk-of-bias assessments in systematic reviews. *International Journal of Epidemiology*, 45(1):266–277, 2016.

[91] André Elisseeff and Jason Weston. A kernel method for multi-labelled classification. In *Proceedings of the Conference on Neural Information Processing Systems*, pages 681–687, 2001.

[92] Johannes Fürnkranz, Eyke Hüllermeier, Eneldo Loza Mencía, and Klaus Brinker. Multilabel classification via calibrated label ranking. *Machine Learning*, 73(2):133–153, 2008.

[93] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. In *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 254–269, 2009.

[94] Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the International Conference on Machine Learning*, page 104, 2004.

[95] Shuiwang Ji and Jieping Ye. Linear dimensionality reduction for multi-label classification. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1077–1082, 2009.

[96] Yao-Nan Chen and Hsuan-Tien Lin. Feature-aware label space dimension reduction for multi-label classification. In *Proceedings of the Conference on Neural Information Processing Systems*, pages 1529–1537, 2012.

[97] Wei Bi and James T Kwok. Efficient multi-label classification with many labels. In *Proceedings of the International Conference on Machine Learning*, pages 397–405, 2013.

[98] Andrew McCallum. Multi-label text classification with a mixture model trained by em. In *Proceedings of the AAAI Workshop on Text Learning*, pages 1–7, 1999.

[99] Rich Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

[100] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(8):2493–2537, 2011.

[101] Hal Daumé III. Frustratingly easy domain adaptation. *arXiv preprint arXiv:0907.1815*, 2009.

[102] Ye Zhang and Byron Wallace. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*, 2015.

[103] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[104] James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. Explainable prediction of medical codes from clinical text. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1101–1111, 2018.

[105] Ke Liu, Junqiu Wu, Shengwen Peng, Chengxiang Zhai, and Shanfeng Zhu. The fudan-uiuc participation in the bioasq challenge task 2a: The antinomyra system. In *CEUR Workshop Proceedings*, volume 1180, pages 1311–1318, 2014.

[106] Grigorios Tsoumakas, Manos Laliotis, Nikos Markantonatos, and Ioannis Vlahavas. Large-scale semantic indexing of biomedical publications at BioASQ. In *BioASQ Workshop*, 2013.

[107] André Elisseeff and Jason Weston. A kernel method for multi-labelled classification. In *Proceedings of the Conference on Neural Information Processing Systems*, pages 681–687, 2002.

[108] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. *Machine Learning*, 85(3), 2011.

[109] Kush Bhatia, Himanshu Jain, Purushottam Kar, Manik Varma, and Prateek Jain. Sparse local embeddings for extreme multi-label classification. In *Proceedings of the Conference on Neural Information Processing Systems*, pages 730–738, 2015.

[110] Hal Daumé III, Nikos Karampatziakis, John Langford, and Paul Mineiro. Logarithmic time one-against-some. In *Proceedings of the International Conference on Machine Learning*, pages 923–932, 2017.

[111] Sheng Chen, Akshay Soni, Aasish Pappu, and Yashar Mehdad. Doctag2vec: An embedding based multi-label learning approach for document tagging. In *Proceedings of the Workshop on Representation Learning for NLP at Annual Meeting of Association for Computational Linguistics*, pages 111–120, 2017.

[112] Yacine Jernite, Anna Choromanska, and David Sontag. Simultaneous learning of trees and representations for extreme classification, with application to language modeling. *arXiv preprint arXiv:1610.04658*, 2016.

[113] Alina Beygelzimer, John Langford, Yuri Lifshits, Gregory Sorkin, and Alex Strehl. Conditional probability tree estimation analysis and algorithms. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 51–58, 2009.

[114] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[115] Gaurav Singh, Iain J Marshall, James Thomas, John Shawe-Taylor, and Byron C Wallace. A neural candidate-selector architecture for automatic structured clinical text annotation. In *Proceedings of the ACM Conference on Information and Knowledge Management*, pages 1519–1528, 2017.

[116] Tal Baumel, Jumana Nassour-Kassis, Raphael Cohen, Michael Elhadad, and Noémie Elhadad. Multi-label classification of patient notes: case study on icd code assignment. In *Workshops at the AAAI Conference on Artificial Intelligence*, 2018.

[117] Zafer Barutcuoglu, Robert E Schapire, and Olga G Troyanskaya. Hierarchical multi-label prediction of gene function. *Bioinformatics*, 22(7):830–836, 2006.

[118] Richard Socher, Brody Huval, Bharath Bath, Christopher D Manning, and Andrew Y Ng. Convolutional-recursive deep learning for 3d object classification. In *Proceedings of the Conference on Neural Information Processing Systems*, pages 656–664, 2012.

[119] Piotr Mirowski and Andreas Vlachos. Dependency recurrent neural language models for sentence completion. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*, pages 511–517, 2015.

[120] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[121] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[122] Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1(2), 1989.

[123] Anthony Rios and Ramakanth Kavuluru. Convolutional neural networks for biomedical text classification: application in indexing biomedical articles. In *Proceedings of the ACM Conference on Bioinformatics, Computational Biology and Health Informatics*, 2015.

[124] Stefanie Nowak and Stefan Rüger. How reliable are annotations via crowdsourcing: a study about inter-annotator agreement for multi-label image annotation. In *Proceedings of the International Conference on Multimedia Information Retrieval*, pages 557–566, 2010.

[125] Ke Liu, Shengwen Peng, Junqiu Wu, Chengxiang Zhai, Hiroshi Mamitsuka, and Shanfeng Zhu. Meshlabeler: improving the accuracy of large-scale mesh indexing by integrating diverse evidence. *Bioinformatics*, 31(12):i339–i347, 2015.

[126] Adler Perotte, Rimma Pivovarov, Karthik Natarajan, Nicole Weiskopf, Frank Wood, and Noémie Elhadad. Diagnosis code assignment: models and evaluation metrics. *Journal of the American Medical Informatics Association*, 21(2), 2013.

[127] Makoto Miwa and Mohit Bansal. End-to-end relation extraction using lstms on sequences and tree structures. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 1105–1116, 2016.

[128] Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. Joint extraction of entities and relations based on a novel tagging scheme. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 1227–1236, 2017.

[129] Heike Adel and Hinrich Schütze. Global normalization of convolutional neural networks for joint entity and relation classification. In *Proceedings of*

the Conference on Empirical Methods in Natural Language Processing, pages 1723–1729, 2017.

[130] Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. A walk-based model on entity graphs for relation extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 81–88, 2018.

[131] Yu Su, Honglei Liu, Semih Yavuz, Izzeddin Gur, Huan Sun, and Xifeng Yan. Global relation embedding for relation extraction. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 820–830, 2018.

[132] Nguyen Bach and Sameer Badaskar. A survey on relation extraction. *Language Technologies Institute, Carnegie Mellon University*, 2007.

[133] Nanda Kambhatla. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the Interactive Poster and Demonstration Sessions at Annual Meeting of Association for Computational Linguistics*, page 22, 2004.

[134] Shantanu Kumar. A survey of deep learning methods for relation extraction. *arXiv preprint arXiv:1705.03645*, 2017.

[135] Katrin Fundel, Robert Küffner, and Ralf Zimmer. Relation extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371, 2006.

[136] Razvan C Bunescu and Raymond J Mooney. A shortest path dependency kernel for relation extraction. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 724–731, 2005.

[137] Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. Kernel methods for relation extraction. *Journal of Machine Learning Research*, 3(2):1083–1106, 2003.

[138] Parminder Bhatia, Busra Celikkaya, and Mohammed Khalilia. End-to-end joint entity extraction and negation detection for clinical text. *arXiv preprint arXiv:1812.05270*, 2018.

[139] Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the Annual Meeting of the ACL and the International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, 2009.

[140] Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1753–1762, 2015.

[141] Bonan Min, Ralph Grishman, Li Wan, Chang Wang, and David Gondek. Distant supervision for relation extraction with an incomplete knowledge base. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 777–782, 2013.

[142] Meishan Zhang, Yue Zhang, and Guohong Fu. End-to-end neural relation extraction with global optimization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1730–1740, 2017.

[143] Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen-tau Yih. Cross-sentence n-ary relation extraction with graph lstms. *Transactions of the Association for Computational Linguistics*, 5(1):101–115, 2017.

[144] Chris Quirk and Hoifung Poon. Distant supervision for relation extraction beyond the sentence boundary. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*, pages 1171–1182, 2017.

[145] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[146] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. *arXiv preprint arXiv:1705.03122*, 2017.

[147] Özlem Uzuner, Brett R South, Shuying Shen, and Scott L DuVall. 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association*, 18(5), 2011.

[148] Jiao Li, Yueping Sun, Robin J Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Thomas C Wiegers, and Zhiyong Lu. Biocreative v cdr task corpus: A resource for chemical disease relation extraction. *Database: The Journal of Biological Databases and Curation*, page 10, 2016.

[149] Chih-Hsuan Wei, Yifan Peng, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Jiao Li, Thomas C Wiegers, and Zhiyong Lu. Assessing the state of the art in biomedical relation extraction: Overview of the biocreative v chemical-disease relation task. *Database: The Journal of Biological Databases & Curation*, page 8, 2016.

[150] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 464–468, 2018.