THE UNIVERSITY OF CHICAGO

ON LEARNING AND APPLYING TEXT REPRESENTATIONS

A DISSERTATION SUBMITTED TO
THE FACULTY OF THE DIVISION OF THE PHYSICAL SCIENCES
IN CANDIDACY FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE

BY

ZEWEI CHU

CHICAGO, ILLINOIS
AUGUST 2020

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to acknowledge my advisor Kevin Gimpel, my thesis committee members (Risi Kondor, Kevin Gimpel, Michael Maire) and all my collaborators (Mingda Chen, Karl Stratos, Hai Wang, David McAllester, Yang Chen, Miaosen Wang, Manaal Faruqui, Robert L. Logan IV, Jun Seok Kang, Dheeru Dua, Sameer Singh, Niranjan Balasubramanian, Jing Chen, Xiance Si) for their guidance, help and contributions during my PhD.

ABSTRACT

Unsupervised learning text representations aims at converting natural languages into vector representations. These vector representations are used in bigger models such as neural networks to improve the performances of supervised tasks. In this line of work, we have Word2Vec [145], Skip-thought [98], ELMo [173], BERT [47], and other improved BERT models such as RoBERTa [124] and ALBERT [102].

To evaluate the effectiveness of these unsupervised learned text representations, people create suites of natural language processing tasks, including SentEval [40] and GLUE [213]. These tasks aims to evaluate the capabilities of these text representations at improving a variety of NLP tasks, including text classification, semantic relatedness and similarity, question answering, sequence labeling, etc.

This thesis discuss our work on both sides. We develop methods to train better language representations and also develop better NLP task suites to evaluate these representations. Most of our pretrained unsupervised models use free text resources available online as training data. We use text and their categories to improve text classification tasks. We use Wikipedia category hierarchies to improve natural language inference tasks. We use Wikipedia document structures to learn sentence representations with discourse information. We also use the hyperlink structures from Wikipedia to learn entity representations. Along with these work we also propose a variety of test suites with standardized tasks to evaluate text representations in these aspects.

CHAPTER 1

INTRODUCTION

1.1 Pretrained Language Representations and Evaluations

Unsupervised learning text representations aims at converting natural languages into vector representations. Word2Vec [145], GloVe [166], Skip-thought [98], ELMo [173], BERT [47], GPT [180], GPT2 [181] are all pretrained models that convert text into vector representations. These vector representations are used in bigger models such as neural networks to improve the performances of supervised tasks. To evaluate the effectiveness of these unsupervised learned text representations, people create suites of natural language processing tasks. These tasks aims to evaluate the capabilities of these text representations at improving a variety of NLP tasks, including text classification, semantic relatedness and similarity, question answering, sequence labeling, etc.

This thesis describes our work on both pretrained text representations and evaluation approachs. We develop methods to train better language representations and also develop better NLP task suites to evaluate these representations.

We find Wikipedia to be a great resource for training unsupervised language representations, as it is freely available to the public, and also comes with different forms of knowledge. Most of our pretrained unsupervised models use Wikipedia as training data.

1.2 Adding Knowledge to Text Representations

Pretrained models such as BERT [47] are shown to already include knowledge [20, 22] by themselves. In this thesis, we work on methods of directly adding knowledge into text representations.

In particular, we explore methods to add knowledge of topic classification, natural language inference, entity information and discourse structure into text representations in this

thesis.

1.3 Evaluation of Text Representations

Vector representations of text are not for humans to comprehend directly, but are useful as model inputs to solve a variety of NLP tasks. To evaluate the usefulness of such text representations, a typical settings is to feed them into a standard NLP test suite. SentEval [40], GLUE [215] and SuperGLUE [212] are popular benchmark datasets for text understanding. Pretrained text representations are often times evaluated on such benchmark test suites to prove their effectiveness.

In this thesis, we propose CatEval, DiscoEval, EntEval along with other evaluation datasets and tasks to evaluate knowledge in text representations.

1.4 Contributions of the Thesis

The key claims of the thesis follow.

- We develop a variety of approaches for learning text representations. By exploring the rich structure of Wikipedia and other text resources, various aspects of text is injected into such representations.
- We propose standard test suites to evaluate text representations, focusing on entity related tasks, discourse related tasks, classification tasks, and language inference tasks.

1.5 Organization of the Thesis

The thesis proceeds as follows.

- Chapter 2 reviews the recent progress on learning text representations, including word embedding, sentence embedding, and contextualized word embedding. This chapter

also discusses approaches of adding different knowledge into such text representations, such as discourse knowledge, entity information, category information and language inferences. While this chapter does not contain novel material, it is a useful resource to review the related work in the field of text representations.

- Chapter 3 describes our work on building weakly supervised text classifiers with naturally annotated text as training resources.

- Chapter 4 describes our work on pretraining text representations for language inference tasks. We use Wikipedia category pairs of parent-child relationship as training resource.

- Chapter 5 describes our work on building entity representations from Wikipedia documents and hyperlinks in them. We also propose a standard benchmark suite EntEval to evaluate entity embedding. This chapter contains material originally published in [30].

- Chapter 6 describes approaches of building discourse knowledge injected sentence representations. The training data is constructed from Wikipedia. We also build a standard test suite DiscoEval to test the effectiveness of different sentence embedding. This chapter contains material originally published in [28].

- Chapter 7 summarizes the contributions of this thesis and discusses the future research directions. In particular, our experience in learning text representations incorporating a variety of knowledge from Wikipedia and its document structures can encourage the explorations of vector representations of different knowledge.

CHAPTER 2

LEARNING TEXT REPRESENTATIONS

In this chapter, we review the current progress in text representation learning. This chapter does not include novel material, but introduce the knowledge background related to this thesis.

2.1 Word and Sentence Embedding

Representing words as vectors is the first step in most deep learning models. A crucial component of this thesis is to find extra knowledge to be encoded into text representations.

The early attempts are word vectors that convert single words into vector representations, such as Word2Vec [145] and GloVe [166]. The word vectors are mostly evaluated on semantic word similarities. Some empirical study also suggest that word vecotrs have nice algebraic properties in the vector space, a famous example being v(king) - v(queen) = v(man) - v(woman), where v(w) is the vector representation of word w.

Inspired by word vectors, sentences vectors are also explored. The goal is to convert natural language sentences into vectors. Among them we have Skip-thought [98] and InferSent [41]. As sentences are more complicated than single words, there are more aspects we can evaluate on these sentence representations. SentEval [40] is proposed as a standardized task suite to evaluate sentence representations. It includes 17 NLP tasks covering topic and sentiment classifications, natural language inferences, semantic similarities and so on.

Both Word2Vec and Skip-thought were trained on the hypothesis of distributional semantics, that "a word/sentence is characterized by the company it keeps". Although the word/sentence vectors are trained by predicting their surrounding words/sentences, in downstream tasks we only use the standalone word/sentence itself.

CoVe [139] and ELMo [173] bring in the idea of contextualized word vectors, where a

word together with its context is encoded together. ELMo is a simple two layer bidirectional LSTM [75] language model trained on large training corpus. Hence the hidden states of each word contains contextual information. The transformer [209] model is a new revolution to this field. The transformer model drops the recurrent operation at training time, thus making training parallelism possible in NLP models. GPT [180] and GPT2 [181] are language models built from the transformer decoder. The ELMo and GPT models outperformed quite a lot of state-of-the-art models on many NLP tasks by the time they were introduced.

The new game changer is BERT [47]. It is trained on the task of masked language modeling, where some words in text are masked and the model is trained to predict these masked words. BERT further improves from ELMo and GPT by large margin on many downstream tasks. Various BERT model improvements are introduces afterwards. RoBERTa is trained with larger data and longer time, while ALBERT shares the parameters across layers but increase hidden size of each layer.

GPT and BERT models are both evaluated on the GLUE [213] benchmark. GLUE includes multiple language understanding tasks including natural language inferences, semantic relatedness and text classifications. Some other tasks are also used as standard evaluation methods for pretrained text mdoels, such as SQuAD [183] for question answering, named entity recognition for sequence labeling, etc.

The above pretrained text representations are shown to be helpful in a variety of NLP tasks. There are two common ways of using such pretrained text representations, feature extraction and fine-tuning. In feature extraction, the text is encoded to be vectors and kept unchanged in later training steps. In fine-tuning, the text encoding layer is treated as a part of the model to be trained, so it is fine-tuned for the specific task. Depending on what the pretrained and downstream tasks are, either feature extraction or fine-tuning may perform better.

2.2 Loss Functions

A variety of loss functions are used in pretraining language representations. We give an overview in this section.

Sentence Decoding Loss The skip-thought model is trained on decoding surrounding sentences from centering sentences. In particular, the encoder is a GRU [35] that encodes the centering sentence to a vector. Conditioned on this vector, two decoders generate the previous and the next sentences in an auto regressive way.

$$\begin{aligned} l(\mathbf{s}_{t-1}, \mathbf{s}_t, \mathbf{s}_{t+1}) = &- \sum_{i=1}^{|\mathbf{s}_{t-1}|} \log p(\mathbf{s}_{t-1,i} | \mathbf{s}_t, \mathbf{s}_{t-1,1}, \cdots, \mathbf{s}_{t-1,i-1}) \\ &- \sum_{i=1}^{|\mathbf{s}_{t+1}|} \log p(\mathbf{s}_{t+1,i} | \mathbf{s}_t, \mathbf{s}_{t+1,1}, \cdots, \mathbf{s}_{t+1,i-1}) \end{aligned}$$

FastSent [73] uses bag-of-words sentence representations for both the encoder and decoder. More specifically, given the centering sentence representation s_i, it is trained to predict the words of surrounding sentences by the following loss:

$$\sum_{w \in S_{i-1} \cup S_{i+1}} \phi(\mathbf{s}_i, v_w) \tag{2.1}$$

Language Modeling Loss ELMo [173] is trained with bidirectional language modeling loss $l_{\text{lang}}(x_{1:T_x}) + l_{\text{lang}}(y_{1:T_y})$ in ELMo where

$$\begin{aligned} l_{\text{lang}}(u_{1:T}) = -\sum_{t=1}^{T} &\log p(u_{t+1} | u_1, \ldots, u_t) + \\ &\log p(u_{t-1} | u_t, \ldots, u_T) \end{aligned}$$

and p is defined by the ELMo parameters.

Masked Language Modeling Loss BERT [47] is trained to reconstruct masked words from the whole text sequence.

$$l_{\text{MLM}}(w_{1:T}) = -\sum_{t \in M} \log p(w_t | \text{mask}(w_1, \cdots, w_T))$$

where M is the set of tokens masked by the [MASK] token.

2.3 Evaluating Text Representations

Vector representations of text are not for humans to read directly, but are useful as model inputs to solve a variety of NLP tasks. To evaluate the usefulness of such text representations, a typical settings is to feed them into a standard NLP test suite.

SentEval [40] is a task suite designed for evaluating sentence representations. Encoded sentence representations as vectors are fed into simple one layer neural network for various tasks, including text classification, sentence similarity, etc.

GLUE [215] and SuperGLUE [212] are two popular benchmarks for language understanding tasks. They do not put restrictions on model design, but only encourage researchers to perform better on their proposed set of tasks. The tasks are focused on one or two sentences.

Recent work has sought to evaluate the knowledge acquired by pretrained language models [188, 1, 14, 172, 42, 40, 214, 121, 28].

2.4 Weakly Supervised Text Classifications

Chapter 3 describes our work on training weakly supervised text classification models with articles and categories from Wikipedia.

There is a great deal of prior work in weakly supervised text classification. A classical

work is the dataless text classification approach approach by [25], later extended to exploit hierarchical label structures by [196]. This approach uses Explicit Semantic Analysis (ESA) [59], a method to represent a document and a candidate category as sparse binary indicators of Wikipedia concepts and compute their relatedness by cosine similarity. [217] propose to learn a universal text classifier based on Wikipedia by extending the dataless approach, but both ESA and their work are pre-neural and outperformed by our models in experiments.

There are many relevant modern approaches to weakly supervised text classification based on neural networks, but they differ from our setting in significant ways. [148] and [185] focus on medical text rather than aiming to handle general open-domain topics which is our goal. [242] propose to enhance the performance of weakly supervised classification by modeling semantic knowledge in the form of class hierarchies and knowledge graphs, but this approach requires nontrivial resources and is difficult to scale. In contrast, we propose to leverage a rich and readily available resource. [143] assume weak supervision in the form of having class keywords and propose a training method by generating pseudo documents based on the keywords and using the documents to train a classifier. This approach is sensitive to the way pseudo documents are generated and rather complex. In contrast, our aim is to show that it is possible to achieve excellent performance with a straightforward approach with simple models.

We briefly mention other related works on weakly supervised classification. [115] also jointly learn label and entity embeddings to perform weakly supervised classification, but they do not learn general text embeddings as in this work. Similarly, [133] focus on learning label embedding to classify named entities. [238] report weakly supervised text classification experiments with a neural model that jointly embeds words and labels, but their experiments are small-scale and do not aim to address the practical setting of handling a wide range of open-domain topics as in this work.

There are other settings of weakly supervised text classification considered in the literature. [45] map both class labels and text into the same semantic space and classify a document by its nearest neighbor of classes in that semantic space. [150] learn the embeddings of all classes, documents, and words together and perform classification. [189] and [55] introduce the problem of Open Domain Classfication (DOC), where a document may belong to a special *unseen* class at test time. To accommodate this special unseen class, they propose to train a 1-vs-rest classifier for each seen category. At test time, an example will be "rejected" if it is not classified into any of the seen categories. [242] perform weakly supervised text classification by a two-phase method. Phase 1 performs a binary classification to decide whether a document belongs to seen categories. The second phase classifies a document to an exact category. This thesis focuses on a more practical setting of using freely available Wikipedia documents and category labels to build a general purpose document topic classifier.

There is also a wealth of prior work in semi-supervised text classification: using unlabeled text to improve classification performance [156]. These methods typically learn generally useful text representations from a large corpus of unlabeled text and use them for a specific target task with limited supervision [79, 173].

Finally, supervised text classification is a well studied problem. A typical approach is to convert text into a vector representation (e.g., bag-of-n-grams) and apply standard classification models such as naive Bayes and support vector machine [218, 89]. Recent works based on neural networks achieve state-of-the-art performance [97, 243, 92, 203, 90, 91]. In particular, attention mechanisms and joint document-label embeddings have been shown to be useful [233, 216].

2.5 Natural Language Inference

Chapter 4 describes our work on pretraining models for natural language inference tasks.

We build on a rich body of literature on leveraging specialized resources (such as knowledge bases) to enhance model performance. These works either (1) pretrain the model on datasets extracted from such resources, or (2) use the resources directly by changing the model itself.

The first approach aims to improve performance at test time by designing useful signals for pretraining, for instance using hyperlinks [127, 27] or document structures in Wikipedia [29], knowledge bases [126], and discourse markers [154]. Here, we focus on using category hierarchies in Wikipedia. There are some previous works that also use category relations derived from knowledge bases [190, 184], but they are used in a particular form of distant supervision in which they are matched with an additional corpus to create noisy labels. In contrast, we use the category relations directly without requiring such additional steps.

Within this first approach, there have been many efforts aimed at harvesting inference rules from raw text [116, 201, 17, 200, 235, 10, 16]. Since WikiNLI uses category pairs in which one is a hyponym of the other, it is more closely related to work in extracting hyponym-hypernym pairs from text [71, 193, 194, 164, 141]. However, most of this prior work uses raw text or raw text combined with either annotated data or curated resources like WordNet. WikiNLI, on the other hand, seeks a middle road, striving to find large-scale, naturally-annotated data that can improve performance on NLI tasks.

The second approach aims to enable the model to leverage knowledge resources during prediction, for instance by computing attention weights over lexical relations in WordNet [31] or linking to reference entities in knowledge bases within the Transformer block [174]. While effective, this approach requires nontrivial and domain-specific modifications of the model itself. In contrast, we develop a simple pretraining method to leverage knowledge bases that can likewise improve the performance of already strong baselines such as BERT without requiring such complex model modifications.

There are some additional related works that focus on the category information of

Wikipedia. [152] extract a dataset based on Wikipedia article or category titles as well as the relations between categories and pages ("WikiNet"), but they do not empirically validate the usefulness of the dataset. In a similarly non-empirical vein, [241] analyze the differences between the graphs from WordNet and the ones from Wikipedia categories. Instead, we address the empirical benefits of leveraging the category information in the modern setting of pretrained text representations.

2.6 Entity Representations

Chapter 5 describes our work on learning good entity representations. This section reviews the previous work on learning entity representations.

Entity linking/disambiguation. Entity linking is a fundamental task in information extraction with a wealth of literature [70, 66, 119, 81, 58, 104, 137]. The goal of this task is to map a mention in context to the corresponding entity in a database. A natural approach is to learn entity representations that enable this mapping. Recent works focused on learning a fixed embedding for each entity using Wikipedia hyperlinks [230, 61, 105]. [67] additionally train context and description embeddings jointly, but this mainly aims to improve the quality of the fixed entity embeddings rather than using the context and description embeddings directly; we find that their context and description encoders perform poorly on EntEval tasks.

A closely related concurrent work by [127] jointly encodes a mention in context and an entity description from Wikipedia to perform zero-shot entity linking. In contrast, here we seek to pretrain a general purpose entity representations that can function well either given or not given entity descriptions or mention contexts.

Other entity-related tasks involve entity typing [229, 149, 46, 179, 36, 158, 157] and coreference resolution [50, 227, 107, 223, 96].

Part of EntEval involves evaluating world knowledge about entities, relating them to fact checking [210, 221, 206, 237, 32], and commonsense learning [8, 21, 113, 144, 239, 208, 202, 240, 187, 182]. Another related line of work is to integrate entity-related knowledge into the training of language models [126, 244, 199].

Knowledge in contextualized word representations. Recent work has sought to evaluate the knowledge acquired by such models [188, 1, 14, 42, 40, 121]. In this work, we focus on evaluating their capabilities in modeling entities.

2.7 Sentence Representations

Chapter 6 talks about our work on incorporating discourse information into sentence representations.

Discourse modelling and discourse parsing have a rich history [135, 13, 246, 94, 85, 112, 222, 120, 117], much of it based on recovering linguistic annotations of discourse structure.

Several researchers have defined tasks related to discourse structure, including sentence ordering [33, 129, 43], sentence clustering [219], and disentangling textual threads [52, 53, 132, 142, 88, 101].

There is a great deal of prior work on pretrained representations [106, 98, 72, 225, 140, 60, 169, 128, 47, 204, 232, 124]. Skip-thought vectors form an effective architecture for general-purpose sentence embeddings. The model encodes a sentence to a vector representation, and then predicts the previous and next sentences in the discourse context. Since Skip-thought performs well in downstream evaluation tasks, we use this neighboring-sentence objective as a starting point for our models.

There is also work on incorporating discourse related objectives into the training of sentence representations. [83] propose binary sentence ordering, conjunction prediction (requiring manually-defined conjunction groups), and next sentence prediction. Similarly, [191]

and [154] create training datasets automatically based on discourse relations provided in the Penn Discourse Treebank (PDTB; 118).

Our work differs from prior work in that we propose a general-purpose pretrained sentence embedding evaluation suite that covers multiple aspects of discourse knowledge and we propose novel training signals based on document structure, including sentence position and section titles, without requiring additional human annotation.

2.8 Summary

This chapter reviews the previous work related to this thesis. They cover text representation learning, weakly supervised text classification, natural language inferences, entity representations, and sentence representations.

CHAPTER 3

NATCAT

This chapter describes our work on building general purpose text classification models with naturally available text and category resources.

3.1 Introduction

We propose to use naturally annotated text with categories from online resources, and build models that can perform general text classification tasks without training on data from downstream tasks. Our approach is to train a scoring function that assigns a score to each potential document-category pair, and text classification becomes a ranking task.

The goal of weakly supervised (aka. "dataless") text classification is to classify documents without a priori knowledge of target labels. It has the obvious advantage over standard supervised approaches of being label-agnostic: a single model can be used for different labels without re-training. But the performance of weakly supervised classifiers is often significantly behind that of supervised models, limiting their usefulness.

Previous works focus on restricted domains such as medical text [148, 185], leverage additional information such as semantic knowledge graphs [242], or carefully exploit weak supervision such as class keywords [143] to achieve satisfactory performance. However, they suffer from a limited scope, a need for nontrivial extra supervision that is difficult to obtain in a large amount, and rather complicated methodologies.

Instead, we consider a more practical setting: is there a readily available resource that we can use to obtain a simple model that can robustly handle a wide range of open-domain text classification tasks? Our primary contribution is a new dataset, NatCat, that can be used to train strong text classification models. NatCat is constructed from naturally annotated text from a variety of online resources. In particular, we use Wikipedia, Stack Exchange,

Text	Israeli ambassador calls peace conference idea 'counterproductive'. A broad international peace conference that has reportedly been suggested by Egypt could be "counterproductive" and shouldn't be discussed until after ...
NatCat	*invasions*, *diplomats*, *peace*, *diplomacy*, *environmentalism*, *Egypt*, *patriotism* ...
AGNews	***international***, *sports*, *science technology*, *business*.

Table 3.1: An instance of weakly supervised topic classification from AGNews, the highest scoring categories from NatCat, and ranked AGNews categories (true class in bold).

and Reddit as the source of document-category pairs.

We train models on NatCat to compute similarity between any document-category pair. As a result, they can be used as off-the-shelf classifiers that produce interpretable and relevant Wikipedia topics for any document. They can also be effortlessly ported to a specific topic classification task and categorize documents under the labels of the task. Table 3.1 illustrates the use of our model on a document from AGNews.

To evaluate, we propose CatEval, a standardized benchmark for evaluating weakly supervised text classification with a choice of datasets, label descriptions for each dataset, and baseline results. We show that training on NatCat leads to strong baselines for future work in weakly supervised text classification, and study the impact of NatCat data domain and pretrained model choice on particular tasks in CatEval.

Our work builds on previous approaches to weakly supervised text classification and is also significantly different in various ways. Unlike generic representations such as Explicit Semantic Analysis (ESA) [25, 59], we explicitly introduce a surrogate training task for neural models (scoring document-category pairs) that faithfully approximates the end goal of text classification. Our scale is much larger than the small dataset-specific experiments in [238],

and we do not require additional supervision at test time such as seed words as in topic modeling approaches [111, 34].

3.2 CatEval Tasks

The goal of weakly supervised text classification is to classify documents into any task-specific categories that are not necessarily seen during training. We seek to achieve this goal by exploiting freely available resources such as Wikipedia, Stack Exchange, and Reddit. Even though their label distributions do not exactly match the target distribution in a downstream task, generalization is possible if they are sufficiently similar. This setting is referred to using several different terms, including dataless classification [25], transfer learning [160], distant supervision [146], and weakly-supervised learning [245].

Our goal is to develop methods that are capable of scoring any candidate label for any document. To evaluate a method, we take the test sets of standard document classification tasks, use the method to score each label from the set of possible labels for that task, and return the label with the highest score. Therefore, for a given document classification task, we need to specify the name of each label. The choice of label names can have a large impact on performance. As in prior work [25, 196], we manually choose words corresponding to labels in the downstream tasks. Our models and ESA use the same label names which are provided in the appendix.

As our choice of text classification tasks, we propose CatEval, which comprises a diverse choice of 11 text classification tasks including both topic-related and sentiment-related labels, and contains both single and multi-label classification tasks. For single label topic classification, we have AGNews,[1] DBpedia [108], Yahoo [243], and 20 News Groups [103].

For sentiment classification, we use Emotion [99], SST-2 [195], Yelp-2, and Amazon-2 [243]. Emotion is a fine-grained sentiment classification tasks with labels expressing

1. https://www.di.unipi.it/~gulli/AG_corpus_of_news_articles.html

dataset	# test docs.	# labels	# sents. per doc.	# words per doc.	# words per sent.
Single label topic classification					
AG	7,600	4	1.3	48.8	36.8
DBP	70k	14	2.4	58.7	24.4
YAHOO	60k	10	5.7	115.8	20.3
20NG	7,532	20	15.9	375.4	
Single label sentiment classification					
Emo.	16k	10	1.6	19.5	12.4
SST	1,821	2	1.0	19.2	19.1
Yelp	38k	2	8.4	155.1	18.4
Amz.	400k	2	4.9	95.7	19.5
Multi-label topic classification					
NYT	10k	100	30.0	688.3	22.9
COM.	1,287	28	1.3	13.8	10.5
Sit.	3,525	12	1.8	44.0	24.7

Table 3.2: Statistics of CatEval datasets.

various emotions, while the other three are binary sentiment classification tasks differentiating positive and negative sentiments.

As for multi-label topical classification, we have NYTimes [186], Comment,[2] and Situation [138] datasets. The NYTimes categories have hierarchical structure, but we merely use the category names from the lowest level. We removed newspaper-specific categories that are not topical in nature.[3] Of the remaining 2295 categories, we only use the 100 most frequent categories in our experiments, and randomly sample 1 million documents for the training set, 10k for a dev set, and 10k as a test set.[4]

Table 3.2 summarizes the key statistics of each dataset, including the average number of sentences, average number of words, and average sentence length. They cover a broad range of text classification tasks and can serve as a benchmark for text classifiers.

2. `https://dataturks.com/projects/zhiqiyubupt/comment`

3. *opinion*, *paid death notices*, *front page*, and *op-ed*

4. Train/dev sets are only used for the supervised baselines.

	Wiki.	StackEx.	Reddit
# categories	1,730,447	156	3,000
# documents	2,800,000	2,138,022	7,393,847
avg. # cats. per doc.	86.9	1	1
mode # cats. per doc.	46	1	1

Table 3.3: Statistics of training sets sampled from the NatCat dataset, with three different data sources from Wikipedia, Stack Exchange and Reddit.

3.3 NatCat Dataset

In this section, we describe the creation procedures of the NatCat dataset. NatCat is constructed from three different data sources: Wikipedia, Stack Exchange, and Reddit. The goal is to construct document-category pairs where the document falls into the category.

Wikipedia. We obtained Wikipedia documents from Wikimedia Downloads. Wikipedia page-to-category mappings were generated from Wiki SQL dumps using the "categorylinks" and "page" tables. We removed hidden categories by SQL filtering, which are typically maintenance and tracking categories that are unrelated to the document content. We also removed disambiguation categories. After filtering, there are 5.75M documents with at least one category, and a total of 1.19M unique categories. We preprocessed the Wikipedia articles by removing irrelevant information such as the external links at the end of each article. We then removed Wikipedia documents with fewer than 100 non-stopwords.

Some category names are lengthy and specific, e.g., "Properties of religious function on the National Register of Historic Places in the United States Virgin Islands". These categories are unlikely to be as useful for end users or downstream applications as shorter and more common categories. Therefore, we consider multiple ways of augmenting the given categories with additional categories.

The first way is to use a heuristic method of breaking long category names into shorter ones. We first use stopwords as separators and keep each part of the non-stopword word

sequence as a category name. For each category name of a document, we also run a named entity recognizer [78] to find all named entities in that category name, and add them to the category set of the document. This way we expand the existing category names from Wikipedia. For the example category above, this procedure yields the following categories: "religious function", "the national register of historic places", "properties", "historic places", "the united states virgin islands", "properties of religious function on the national register of historic places in the united states virgin islands", "united states virgin islands", "national register".

Our second method of expansion is based on the fact that Wikipedia categories can have parent categories and therefore form a hierarchical structure. We augment an article's category set by adding categories that are within two edges from any of the initial categories. In expanding the category set in this way, there is a trade-off between specificity/relevance and generality/utility of category names. Using only the categories provided for the article yields a small set of high-precision, specific categories. Adding categories that are one or two edges away in the graph increases the total number of training pairs and targets more general/common categories, but some of them will be less relevant to the article. In NatCat, we include all categories of documents that are up to two edges away.

Stack Exchange. Stack Exchange is a question answering platform where users post and answer questions as a community. Questions on Stack Exchange fall into 308 subareas, each area having its own site. We construct the document-category pair dataset by pairing question titles or descriptions with their corresponding subareas. Question titles, descriptions and subareas are available from [37]. Many Stack Exchange subareas have their own corresponding "meta" sites. When creating this dataset, we merge the subareas with their corresponding "meta" area. This gives us over 2 million documents with 156 categories.

Reddit. Inspired by [178], we construct a category classification dataset from Reddit. In our dataset, we propose to classify Reddit post titles to their corresponding subreddit names. We use the OpenWebText[5] toolkit to get Reddit posts with more than 3 karma and their subreddit names. We only keep the top 3k most frequent subreddits as they better capture the common categories that we are interested in. This gives us over 7 million documents with 3k categories.

Table 3.3 summarizes statistics of training sets we sampled from the NatCat dataset. Note that all documents from Stack Exchange and Reddit only have one associated category, while a document from Wikipedia may have multiple categories describing it.

3.3.1 Relationship to CatEval *Tasks*

As NatCat is a large textual resource with ample categories, almost all labels in the CatEval datasets appear in NatCat except for some conjunction phrases, such as "written work", "manufacturing operations and logistics", and "home and garden". However, there is no guarantee that the labels in NatCat have the same definition as the labels in the downstream tasks, and in fact we find such divergences to be causes of error, including when measuring human performance on the text classification tasks. Weakly supervised methods (and humans) are more susceptible to semantic imprecision in label names than supervised methods. The reason we describe our method as "weakly supervised" is because it does not require annotated training data with the same labeling schema and from the same distribution as the test set, but rather uses freely-available, naturally-annotated document/category pairs as a training resource.

5. `https://github.com/jcpeterson/openwebtext`

3.4 Experiments

3.4.1 Models and Training

We train BERT [48] and RoBERTa [125] on the NatCat dataset for weakly supervised classification. In our experiments, we use BERT-base-uncased (110M parameters) and RoBERTa-base (110M parameters). We formulate NatCat training as a binary classification task to predict whether a category correctly describes a document. For each document-category pair, we randomly sample 7 negative categories for training. As documents from Wikipedia have multiple positive categories, we randomly sample one positive category for each of them.

We concatenate the category with the document as the input for BERT and RoBERTa: "[CLS] category [SEP] document [SEP]". In our experiments, we truncate the document to ensure the category-document pair is within 128 tokens.

For BERT and RoBERTa models shown in Table 3.4 and Table 3.5, we train models on the whole NatCat dataset and also the data from a single resource (Wikipedia, Stack Exchange, or Reddit). For each single domain, we train the model for one epoch on 100k instances. For NatCat combining three domains, we train on 300k instances. The learning rate is set to be 0.00002, and we perform learning rate warmup for 10% of the training steps and then linearly decay the learning rate. As BERT and RoBERTa models are known to suffer from randomness among different runs, we perform each single experiment 5 times under different random seeds and report the median of such five runs. We also do supervised training on Emotion, NYTimes, Comment, Situation with the RoBERTa model. We follow the same training procedure as we train on NatCat to solve a document-category binary classification task. Our training code is built on Huggingface Transformers [228] and will be released upon publication.

We compare to ESA, for which we use their provided code.[6] We followed the methods of

6. `github.com/CogComp/cogcomp-nlp/tree/master/dataless-classifier`

dataless classification from [25]. Instead of setting a threshold on the number of concepts as in prior work, we use all Wikipedia concepts as we find this improves ESA's performance.

In preliminary experiments, we experimented with other unsupervised text representation learning approaches, e.g., encode the document and category using pretrained models or pretrained word embeddings, then use cosine similarity as the scoring function for a document-category pair. However, we found these methods do not perform as well as weakly supervised approaches such as ESA and our approach, so we do not report the results of such methods in this paper.

3.4.2 Evaluation

We report classification accuracy for all single label classification tasks, including topical and sentiment tasks. For multi label classification tasks, we use label ranking average precision (LRAP):

$$\text{LRAP}(y, \hat{f}) = \frac{1}{n} \sum_{i=0}^{n-1} \frac{1}{\|y_i\|_0} \sum_{j:y_{ij}=1} \frac{|\ell_{ij}|}{rank_{ij}}$$

where $\hat{f}$ are prediction scores, y are ground truth labels, n is the number of samples, $\ell_{ij} = \{k : y_{ik} = 1, \hat{f}_{ik} \geq \hat{f}_{ij}\}$, and $rank_{ij} = |\{k : \hat{f}_{ik} \geq \hat{f}_{ij}\}|$.

3.4.3 Primary Results

	Topical (Acc)					Sentiment (Acc)				
	AG	DBP	YAH.	20NG	AVG	Emo	SST	Yelp	Amz	AVG
BERT models										
Wikipedia	72.3	86.0	49.0	33.3	60.5	21.3	63.8	64.5	67.0	**66.6**
StackEx.	69.0	76.0	59.1	**51.2**	**64.0**	18.7	60.1	57.8	57.0	59.1
Reddit	70.3	72.8	51.8	49.2	61.7	12.5	61.2	67.0	66.2	65.2
NatCat	**75.6**	82.8	54.9	39.3	63.3	16.1	62.7	70.4	63.6	53.8
NatCat ens.	75.4	83.0	55.2	41.7	63.8	16.6	65.7	67.6	68.4	54.6
RoBERTa models										
Wikipedia	71.7	**87.1**	53.1	38.8	62.6	**22.6**	57.2	66.3	69.7	65.3
StackEx.	65.9	75.5	**59.3**	19.6	54.7	21.7	59.9	66.2	60.8	62.4
Reddit	61.7	71.2	54.0	10.4	49.5	21.3	59.5	57.2	62.9	61.1
NatCat	68.8	81.9	57.8	36.8	61.3	21.2	65.0	67.3	66.8	55.8
NatCat ens.	68.4	85.0	58.5	37.6	62.4	22.3	**68.7**	**75.2**	72.4	59.7
Other weakly supervised models										
ESA	71.2	62.5	29.7	25.1	47.1	9.5	52.1	51.1	51.9	41.2
Yin et al.	-	-	52.1	-	-	21.2	-	-	-	
PC19 1/4	68.3	52.5	52.2	-	-	-	61.7	58.5	64.5	
PC19 All	65.5	44.8	49.5	-	-	-	62.5	74.7	**80.2**	
Fully supervised and human performances										
Supervised	92.4	98.7	71.2	85.5	87.0	34.5	75.3	95.6	95.1	75.1
Human	83.8	88.2	75.0	-	-	-	-	-	-	

Table 3.4: Results of BERT and RoBERTa trained on NatCat and evaluated on Cat-Eval. Results are shown for training on both the full NatCat dataset as well as individual NatCat data sources. NatCat ens. is an ensemble over NatCat-trained models with 5 random seeds. We compare with the reported zero-shot results from [236] and [178]. We also compare with results from supervised methods. The supervised results of AGNews, DBpedia, Yahoo, Yelp-2 and Amazon-2 are from [243]. The SST-2 result is from [220]. The 20 News Groups result is from [163]. PC19 results [178] are GPT2 medium models fine-tuned on 1/4 and all of their training data.

	Multi label topical (LRAP)				
	NYT	COM.	Sit.	AVG	All
BERT models					
Wikipedia	41.8	24.3	51.1	39.0	53.0
StackEx.	36.5	24.1	49.9	36.8	51.0
Reddit	49.8	22.6	52.4	**41.5**	52.1
NatCat	49.6	22.6	50.5	41.0	53.3
NatCat ens.	**50.8**	22.6	50.8	41.4	54.3
RoBERTa models					
Wikipedia	37.9	23.1	49.9	37.1	52.7
StackEx.	37.7	**24.6**	47.9	36.8	49.4
Reddit	42.4	20.6	48.4	37.1	47.1
NatCat	47.7	21.5	52.3	40.5	53.4
NatCat ens.	49.0	22.1	52.6	41.2	**55.6**
Other weakly supervised models					
ESA	10.9	22.5	**55.6**	29.7	40.2
Fully supervised and human performances					
Supervised	72.5	64.7	75.2	70.8	63.8

Table 3.5: Results of BERT and RoBERTa trained on NatCat and evaluated on CatEval multi label topical classification tasks. Results are shown for training on both the full NatCat dataset as well as individual NatCat data sources. NatCat ens. is an ensemble over NatCat-trained models with 5 random seeds. We compare with the reported zero-shot results from [236] and [178]. We also compare with results from supervised methods. The supervised results of NYTimes, Situation, Comment and Emotion results are fine-tuned RoBERTa models.

Table 3.4 and Table 3.5 summarizes the experimental results of BERT and RoBERTa models trained on NatCat and evaluated on CatEval. RoBERTa trained on NatCat performs the best on average across tasks, but there are some differences between BERT and RoBERTa. BERT is better on AGNews and NYTimes, both of which are in the newswire domain, as well as 20NG, which also involves some news- or technical-related material. RoBERTa is better on Yahoo as well as better on average in the emotion, binary sentiment, and situation tasks. This may be due to RoBERTa's greater diversity of training data (web text) compared to BERT's use of Wikipedia and books.

Models trained on Stack Exchange do not perform well on most sentiment related tasks. This is likely because Stack Exchange subareas are divided by topic. Wikipedia and Reddit are better resources for training sentiment classifiers, as they cover broader ranges of sentiment and emotion knowledge.

All data from three different resources are good at some particular topical classification tasks, most of which can be explained by domain similarities. For example, models trained on Wikipedia are good at DBpedia, which can be explained by the fact that DBpedia is also built from Wikipedia. Stack Exchange is especially helpful for Yahoo; both are in the domain of community question answering. Models trained on Reddit, which contains a sizable amount of political commentary and news discussion in its most frequent categories, are particularly good at NYTimes.

Compared to other weakly supervised methods [236, 178], NatCat trained models have the advantage on all topical classification tasks.

To provide perspective on the difficulty of the weakly supervised setting, we obtained annotations from human annotators involved in this research project on 60 instances from AGNews, 50 from DBpedia, and 100 from Yahoo. We showed annotators instances and the set of class labels and asked them to choose a single category without the ability to look at any training examples.

	Wiki	StackEx	Reddit	NatCat	Yin et al.
BERT	23.4	38.6	32.5	37.1	27.7
RoBERTa	26.7	36.2	34.6	36.2	-

Table 3.6: Results (F1 scores) for the Situation task. While Table 3.5 reports LRAP, here we show F1 in order to compare to Yin et al.

	YAHOO		Emotion		Situation	
	seen	unseen	seen	unseen	seen	unseen
BERT	73.2	12.9	32.8	17.8	73.1	50.4
+ NatCat	73.6	16.5	33.4	17.6	72.8	48.6
Yin et al.	*72.6*	*44.3*	35.6	17.5	72.4	48.4

Table 3.7: Results on half seen text classification tasks.

In some tasks (AGNews and DBpedia), supervised models outperform human annotators. We believe this is caused by semantic drift between human interpretation and the actual meaning of the labels as determined in the dataset. Supervised models are capable of learning such nuance from the training data, while an annotator without training is not capable of classifying documents in that way. Weakly supervised models are like human annotators in that they are only capable of classifying documents with the general knowledge they have learned (in this case from large scale naturally-annotated document-category resources).

In order to directly compare to the results from [236] for multi-label classification, we also report the label-weighted F1 scores of the Situation task in Table 3.6. In comparing with the Wikipedia-based training from [236], NatCat-trained models are better at the situation detection task.

3.5 Half Seen Settings

Zero-shot text classification is often defined as training models on some seen labels and testing on an expanded set of both seen and unseen labels [238, 236].

We follow the same seen and unseen label splits as [236], using their v0 splits. We use the same parameter setting as Section 3.4, train BERT models for 3 epochs on the seen label sets, and predict over both seen and unseen labels. We train both the original BERT-base-uncased model and the NatCat-pretrained BERT model from Section 3.4. Table 3.7 summarizes the results (medians over 5 random seeds).[7] The evaluation metrics are accuracy for Yahoo and label-weighted F1 for Emotion and Situation, in order to compare to Yin et al. Pretraining on NatCat improves BERT's results on Yahoo, but it does not show clear improvements on Emotion and Situation in this setting.

Our Yahoo results are not directly comparable to the results from [236] for several reasons, the most significant being that Yin et al. expand label names using their definitions in WordNet, while we choose to use the plain label names for our experiments.[8] Another important difference is that [236] implement a "harsh policy" to impose an advantage to unseen labels by adding an α value to the probabilities of unseen labels. This α value is set by tuning on the development set which contains both seen and unseen labels. However, we do not assume access to a development set with unseen labels.

The unseen label results on these tasks are generally not as good as NatCat trained models from Table 3.4 and Table 3.5, making the weakly supervised approach more appealing than this half-seen setting. This discrepancy may be due to some "catastrophic forgetting" in the final stage of training.

7. The Yahoo dataset used in Table 3.7 is from [236], which is slightly different from that of [243] used in Table 3.4 and Table 3.5.

8. Also, Yin et al. formulate the problem as an entailment task, and there are differences in training set sizes.

	Topic	Senti.	Multi-label topic	All
Wiki.	1.1/0.6	3.1/2.8	0.5/0.3	1.3/1.2
StackEx.	0.8/1.2	0.7/2.6	0.5/0.7	0.2/1.2
Reddit	0.8/1.5	3.4/1.8	0.3/1.2	1.4/1.4
NatCat	0.8/1.2	3.6/1.8	0.7/0.4	1.3/0.2

Table 3.8: Standard deviations of BERT and RoBERTa model performances on CatEval tasks with 5 different random seeds.

3.6 Analysis

3.6.1 Training Sizes

While NatCat has over 10 million documents with over a million categories, we have used a small subset of it due to computational constraints. We here compare models trained on 100k and 300k document-category pairs, following the same hyperparameter settings as in Section 3.4. We find that increasing training size generally harms performance on CatEval tasks. For example, averaging over all CatEval tasks, BERT trained on Wikipedia is 1.5 points lower when moving from 100k training instances to 300k instances. For Stack Exchange, the gap is 2.1 points. For Reddit, it is 0.7 points.

This is likely due to overfitting on the NatCat binary classification tasks. As there is a discrepancy between training and evaluation, increasing training data or epochs may not necessarily improve results on downstream tasks. This is a general phenomenon in weakly supervised and zero-shot classification, as we do not have development sets to tune training parameters for such tasks. Similar findings were reported by [178], suggesting future work to figure out good ways to do model selection in zero-shot settings.

3.6.2 Model Variances

BERT and RoBERTa are known to suffer from instability in fine-tuning, i.e., training with different random seeds may yield models with vastly different results. To study this phe-

	Topical (Acc)				Sentiment (Acc)			
	AG	DBP	YAH.	20NG	Emo	SST	Yelp	Amz
GPT2 models without candidate answers								
S	55.8	43.7	31.1	25.1	14.1	66.7	66.4	70.0
M	56.4	35.3	32.7	**28.1**	17.3	66.2	69.5	71.8
L	51.1	42.6	36.7	21.8	**17.7**	60.4	65.8	69.5
S+NC	51.5	34.2	29.7	14.5	10.2	66.6	68.6	71.1
M+NC	49.9	42.2	28.2	13.0	11.5	53.2	61.5	58.6
GPT2 models with candidate answers								
M	37	7.7	9.9	4.2	5.5	53.9	58.8	60.7
M+NC	**72.5**	**72.6**	28.4	4.2	14.9	57.7	60.2	63.7
[178]								
1/4	68.3	52.5	**52.2**	-	-	61.7	58.5	64.5
All	65.5	44.8	49.5	-	-	**62.5**	**74.7**	**80.2**

Table 3.9: GPT2 results. S/M/L are small/medium/large pretrained GPT2 models, and models with "+NC" fine-tune GPT2 on NatCat.

nomenon in our setting, we performed training in each setting with 5 random seeds and calculate standard deviations for different tasks. As shown in table 3.8, both models have higher variance on sentiment tasks compared to topic classification. While nontrivial variances are observed, ensembling the 5 models almost always outperforms the median of the individual models.

3.6.3 Experiments with GPT2 Models

We also report preliminary results in adapting GPT2 models to perform CatEval tasks. To do so, we construct the following descriptive text: "The document is about [category]: [document content]", where [category] is replaced by the class label we want to score, and [document content] is the document we want to classify. The descriptive text is tokenized by the BPE tokenizer, truncated to 256 tokens, and fed into the pretrained GPT2 model. The class label with the lowest average loss over all tokens is picked as the predicted label.

The results are shown in the initial rows of Table 3.9. We find mixed results across tasks, with the GPT2 models performing well on sentiment tasks but struggling on the topical

tasks. Increasing GPT2 model size helps in some tasks but hurts in others. The GPT2 small model actually outperforms the 1/4-data training setting from [178] on the sentiment tasks, though not the All-data training.

We also fine-tune GPT2 models on NatCat. Each document in NatCat is paired with its category to construct the aforementioned descriptive text, and fine-tuned as a language modeling task.[9] The results (upper section of Table 3.9), are mixed, with the topical accuracies decreasing on average and the sentiment accuracies slightly increasing for GPT2 small but decreasing for GPT2 medium.

A key difference between training GPT2 and BERT/RoBERTa is that with GPT2, we do not explicitly feed information about negative categories. One way to incorporate this information is to construct descriptive text with "candidate categories" following [178][10] We sample 7 negative categories and 1 correct category to form the candidates. The results, shown in the middle section of Table 3.9, improve greatly for some tasks.

Compared to BERT and RoBERTa, it is harder to fine-tune a GPT2 model that performs well across CatEval tasks. In fact, there are many ways to convert text classification into language modeling tasks; we explored two and found dramatically different performance from them. It remains an open question how to best formulate text classification for pretrained language models, and how to fine-tune such models on datasets like NatCat.

3.6.4 Error Analysis

Upon analysis of the confusion matrix of the RoBERTa ensemble predictions on AGNews, DBpedia, and Yahoo, we observe the following common misclassification instances:

- In AGNews, *science & technology* and *international* are often misclassified as *business*.

9. The learning rate (set to 0.00002) follows linear warmup and decay. Following [178], we set 1% of training steps as warmup period. We train for one epoch. The maximum sequence length is 256 tokens. We use batch size 8 for GPT2 small (117M parameters) and 2 for GPT2 medium (345M parameters).

10. The descriptive text is as follows: "`<|question|>` + question + candidate categories + `<|endoftext|>` + `<|text|>` + document + `<|endoftext|>` + `<|answer|>` + correct category + `<|endoftext|>`" .

- In DBpedia, *nature* is often misclassified as *animal*, *nature* as *plant*, *written work* as *artist*, and *company* as *transportation*.
- In Yahoo, *society & culture* is often misclassified as *education & reference*, *politics & government*, and *business & finance*. *health* is often misclassified into *science & mathematics*, *family relationships* as *society culture*.

The RoBERTa model trained on NatCat confuses closely related categories, but it rarely makes mistakes between clearly unrelated concepts. We find that human errors follow the same pattern: they mostly consist of closely related categories. This suggests that models trained on NatCat are effective at classifying documents into coarse-grained categories, but fine-grained categorization may require annotated training data specific to the task of interest.

3.7 Summary

In this chapter, we presented a practical approach to building general-purpose document classifiers by leveraging freely available document-category pairs from online resources. We will release the NatCat dataset, CatEval benchmark dataset, our code for running experiments and for evaluation, and our best pretrained model. Our model not only handles any label set but also supplies a myriad of interpretable categories for a document off-the-shelf. We believe it can be a useful tool for applications in natural language processing, information retrieval, and text mining.

CHAPTER 4

NATURAL LANGUAGE INFERENCE WITH WIKIPEDIA CATEGORY STRUCTURES

This chapter describes our work on mining knowledge for natural language inference from Wikipedia category structures.

4.1 Introduction

Learning concept hierarchies, such as lexical entailment or natural language inference, has been an important area in natural language processing. Researchers typically use external knowledge bases like WordNet [56], FrameNet [9], or Wikidata [211] or resort to large-scale human-annotated datasets [21, 226, 155]. However, acquiring these resources generally requires expensive human annotations. In this work, we are interested in automatically generating a large-scale dataset from Wikipedia categories that can benefit model performance on both NLI and LE tasks.

We take advantage of the naturally-annotated Wikipedia category graph, where we observe that most of the parent-child category pairs are entailment relationships, i.e., a child category entails a parent category. More importantly, compared to WordNet and Wikidata, the Wikipedia category graph has more fine-grained connections, which could be helpful for training models. Inspired by this observation, we construct WikiNLI by automatic filtering from the Wikipedia category graph. The dataset has 433,899 pairs of phrases and contains three categories, each of which corresponds to a relationship in NLI.

To empirically demonstrate the usefulness of WikiNLI, we pretrain BERT and RoBERTa on WikiNLI, WordNet, and Wikidata, before finetuning on various LE and NLI tasks. Our experimental results show that WikiNLI gives the best performance averaging over 10 tasks, and more importantly, the benefit can generalize to multiple models.

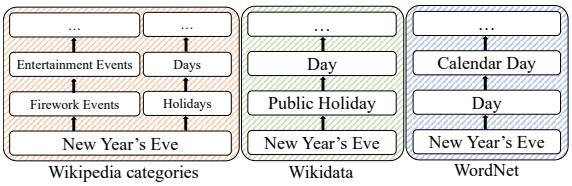


Figure 4.1: Example hierarchies obtained from Wikipedia categories, Wikidata, and WordNet.

We perform an in-depth analysis of approaches to handling the Wikipedia category graph and the effects of pretraining with WikiNLI and other data sources under different configurations. We find that WikiNLI brings consistent improvements in a low resource NLI setting where there are limited amounts of training data, and the improvements plateau as the number of training instances increases; more WikiNLI instances for pretraining are beneficial for downstream finetuning tasks with pretraining on a fourway variant of WikiNLI showing more significant gains for the task requiring higher-level conceptual knowledge; WikiNLI also introduces additional knowledge related to lexical relations benefiting finer-grained LE and NLI tasks; relatively higher levels of knowledge from WikiNLI have more potential of enhancing the performance of NLI systems.

We also construct WikiSentNLI using hyperlinks from Wikipedia for evaluating the effect of including sentential context from Wikipedia category pairs. With a straightforward modification on Wikipedia by including the lowest levels of WikiNLI categories, it achieves promising results.

4.2 WikiNLI

We now describe how the WikiNLI dataset is constructed from Wikipedia and its principal characteristics. Each Wikipedia article is associated with crowd-sourced categories that correspond to topics or concepts covered by that article. Wikipedia organizes these categories into a directed graph that models their hierarchical relations. For instance, the category "Days" is a parent node of the category "Holidays" in this graph. The central observation

underlying WikiNLI is that this category hierarchy resembles the concept hierarchies and ontologies found in knowledge bases, such as Wikidata and WordNet.

While there are similarities between the three resources, the Wikipedia category hierarchy contains more diverse connections between parent and child concepts. Figure 4.1 shows an example category "New Year's Eve" and its ancestors under these resources. All resources include a path that corresponds to the generalization of New Year's Eve as a regular day, but Wikipedia additionally includes a path that corresponds to the generalization as celebration or entertainment. Thus the Wikipedia hierarchy provides more abstract and fine-grained generalization that can be useful for NLI tasks. In this example, the common-sense knowledge that New Year's Eve implies entertainment is only directly captured by the Wikipedia hierarchy.

WikiNLI is a dataset of category pairs extracted from this Wikipedia hierarchy to be used as a useful auxiliary task for pretraining NLI models. Specifically, WikiNLI contains three types of category pairs based on their relations in the Wikipedia hierarchy: child-parent ("child"), parent-child ("parent"), and other pairs ("neutral"). The motivation is that child-parent resembles entailment; parent-child resembles reverse entailment; and other pairs resemble a neutral relationship. We find that this simple definition of relations is effective in practice; we also report an exploration with other types of relations such as siblings in experiments.

Table 4.1 shows examples from WikiNLI that illustrate the diverse set of relations they address. They include conventional knowledge base entries such as "Bone fractures" being a type of "Injuries" and "Chemical accident" being a type of "Polution". They also include relations that are more fine-grained than those typically found in knowledge bases. For instance, "Pakistan" is a child of "South Asian countries"; in contrast, it is a child of "Country" as in Wikidata. They include a large set of hyponym-hypernym relations often in pairs that differ by one or two words (e.g., "Cantonese music" and "Cantonese culture");

Category 1	Category 2	Rel.
Injuries	Bone fractures	P
Chemical accident	Pollution	C
Armenian sportspeople	Curaçao male actors	N
Argentine design	Nigerian inventions	N
Cantonese music	Cantonese culture	C
Medieval Anatolia	Early Turkish Anatolia	P
Learned societies	Academic organizations	C
South Asian countries	Pakistan	P

Table 4.1: Examples from WikiNLI. C = child; P = parent; N = neutral.

their coverage is extensive and includes relations involving rare words such as "Early Turkish Anatolia" and "Medieval Anatolia".

More details of constructing WikiNLI are as follows. We use the tables "categorylinks" and "page": these two pages provide category pairs in which one category is the parent of the other. We use all direct category relations. To eliminate trivial pairs, we remove pairs where either is a substring of the other. To construct neutral pairs, we randomly sample two categories where neither category is the ancestor of the other in the category graph. To make neutral pairs more "related" (so that they are harder to discriminate from direct relations), we encode both categories into continuous vectors using ELMo [170] (averaging its three layers over all positions) and compute the cosine similarities between pairs. We pick the top-ranked pairs as neutral pairs in WikiNLI. After the above processing, we remove categories longer than 50 characters[1] and those containing certain keywords.[2] We ensure the dataset is balanced, and the final dataset has 433,899 pairs.

For the following experiments, unless otherwise specified, we only use 100,000 samples from WikiNLI as training data and 5,000 as the development set due to computational constraints. We will release the full WikiNLI dataset upon publication.

1. We experimented with removing this 50-character limitation but did not see much difference in the experimental results.

2. all digits, ., !, ?, of, at, in, by, from, to, about, stubs, lists.

Dataset	#train	#dev	#test
Natural Language Inference			
MNLI	3000	9815	9796
RTE	2490	277	3000
PPDB	13904	4633	4641
Break	-	-	8193
Lexical Entailment			
K2010	739	82	621
B2012	791	87	536
T2014	539	59	507

Table 4.2: Dataset statistics.

4.3 Approach

To demonstrate the effectiveness of WikiNLI, we pretrain BERT and RoBERTa on WikiNLI and other resources, and then finetune them on several NLI and LE tasks. We assume that if a pretraining resource is better aligned with downstream tasks, it will lead to better downstream performance of the models pretrained on it.

4.3.1 Training

Following [48], we use the concatenation of two texts as the input to BERT. Specifically, for a pair of input texts x_1, x_2, the input would be [CLS]x_1[SEP]x_2[SEP]. We use the encoded representations at the position of [CLS] as the input to a two-layer classifier, and finetune the entire model.

We start with a pretrained BERT-Large or RoBERTa-large model and further pretrain it on different pretraining resources. After that, we finetune the model on the training sets for the downstream tasks, as we will elaborate on below.

4.3.2 Evaluation

Natural Language Inference

MNLI. The Multi-Genre Natural Language Inference (MNLI; 226) dataset is a human-annotated multi-domain NLI dataset. MNLI has three categories: entailment, contradiction, and neutral. Since the training split for this dataset has a large number of instances, models trained on it are capable of picking up information needed regardless of the quality of pretraining resources, which makes the effects of pretraining resources negligible. To better compare the impact of various pretraining resources, we simulate a low-resource scenario by randomly sampling 3,000 instances from the original training split as our new training set, but use the standard "matched" development and testing splits.

RTE. We evaluate models on the GLUE [213] version of the recognizing textual entailment (RTE) dataset [44, 11, 63, 15]. RTE is a binary task, focusing on identifying if a pair of input sentences has the entailment relation.

PPDB. We use the human-annotated phrase pair dataset from [165], which has 9 text pair relationship labels. The labels are: hyponym, hypernym, synonym, antonym, alternation, other-related, NA, independent, and none. We include this dataset for more fine-grained evaluation. Since there is no standard development or testing set for this dataset, we randomly sample 60%/20%/20% as our train/dev/test sets.

Break. [64] constructed a challenging NLI dataset called "Break" using external knowledge bases such as WordNet. Since sentence pairs in the dataset only differ by one or two words, similar to a pair of adversarial examples, it has broken many NLI systems.

Due to the fact that Break does not have a training split, we use the aforementioned subsampled MNLI training set as a training set for this dataset. We select the best performing

	Natural Language Inference				Lexical Entailment			
	MNLI	RTE	PPDB	Break	K2010	B2012	T2014	avg.
BERT	75.0	69.9	66.7	80.2	85.2	79.4	63.3	74.2
+WordNet	75.8	71.3	71.1	83.5	83.5	94.3	71.2	78.7
+Wikidata	75.7	71.3	**75.0**	81.3	82.3	95.3	70.5	78.8
+WikiNLI	76.4	70.9	70.7	**85.7**	84.9	**96.1**	71.2	79.4
RoBERTa	82.5	78.8	65.9	81.3	85.3	65.9	66.8	75.2
+WordNet	83.8	82.2	72.0	82.3	82.5	88.6	70.7	80.3
+Wikidata	84.0	82.3	72.5	83.2	82.4	94.8	71.0	81.5
+WikiNLI	**84.4**	**83.1**	71.7	83.8	**85.4**	95.7	**72.9**	**82.4**

Table 4.3: Test set performance for baselines and models pretrained on various resources. We report accuracy (%) for NLI tasks and F_1 score (%) for LE tasks. The highest results for each model (BERT or RoBERTa) are underlined. The highest numbers in each column are boldfaced.

model on the development set of MNLI and evaluate it on Break.

Lexical Entailment

We use the lexical splits for 3 datasets from [110], including K2010 [100], B2012 [12] and L2014 [109] . These datasets all similarly formulate lexical entailment as a binary task, and they were constructed from diverse sources, including human annotations, WordNet, and Wikidata.

Statistics for these datasets are shown in Table 4.2.

4.4 Experiments

4.4.1 Baselines

We consider three baselines for BERT, namely the original BERT model, BERT pretrained on WordNet, and BERT pretrained on Wikidata. We use one baseline for RoBERTa: the original RoBERTa model.

WordNet. WordNet is a widely-used lexical knowledge base, where words or phrases are connected by several lexical relations. We consider direct hyponym-hypernym relations available from WordNet, resulting in 74,645 pairs.

Wikidata. Wikidata is a database that stores items and relations between these items. Unlike WordNet, Wikidata consists of items beyond word types and commonly seen phrases, offering more diverse domains similar to WikiNLI. The available conceptual relations in Wikidata are: "subclass of" and "instance of". In this work, we consider the "subclass of" relation in Wikidata because (1) it is the most similar relation to category hierarchies from Wikipedia; (2) the relation "instance of" typically involves more detailed information, which is found less useful empirically (see Sec. 4.5.2 for details). The filtered data has 2,871,194 pairs.

We create training sets from both WordNet and Wikidata following the same procedures used to create WikiNLI. All three datasets are constructed from their corresponding parent-child relationship pairs. Neutral pairs are first randomly sampled from non-ancestor-descendant relationships and then keep top ranked pairs by cosine similarities of ELMo embeddings. We also ensure these datasets are balanced among the three classes.

4.4.2 Setup

For all the experiments, we used the Hugging Face implementation [228]. When finetuning or pretraining BERT-Large models, we mostly follow the hyperparameters suggested by [48]. Specifically, during pretraining, we use a batch size of 32, a learning rate of 2e-5, and a maximum sequence length of 40, 3 training epochs, whereas during finetuning we switch to use 8 as batch size due to memory constraints. When funetuning or pretraining RoBERTa-large, we did extra hyperparameter searching by adopting some of hyperparameters recommended from [124]. We use 10% training steps for learning rate warmup, 1e-5 for learning rate, and

a maximum sequence length of 40, and train models for 3 epochs.[3]

For both models, we use development sets for model selection during pretraining. During downstream evaluations, we use a maximum sequence length of 128 for datasets involving sentences. We perform early stopping based on task-specific development sets and report the test results for the best models. Due to the variance of performance of 24-layer transformer architectures, we report medians of 5 runs with a fixed set of random seeds for all of our experiments.

4.4.3 Results

The results are summarized in Table 4.3. We report accuracy (%) for NLI tasks and F_1 score (%) for LE tasks. In general, pretraining on WikiNLI improves the performances on downstream tasks by a significant margin, especially for Break and MNLI, where WikiNLI can lead to much more substantial gains than the other two resources. In some cases, such as RTE and L2014, WordNet or Wikidata may be a better choice of the pretraining dataset. More importantly, the improvements to both BERT and RoBERTa brought by WikiNLI show that the benefit of the WikiNLI dataset can generalize to multiple models.

4.5 Analysis

We perform several kinds of analysis using BERT to compare the effects of different settings.

4.5.1 Fourway vs. Threeway vs. Binary Pretraining

We investigate the effects of the number of categories for WikiNLI by empirically comparing three settings: fourway, threeway, and binary classification. For fourway classification, we add an extra relation "sibling" in addition to child, parent, and neutral relationships. A

3. We choose this set of hyperparameters due to computational constraints. Our finetuned RoBERTa achieves 82.3% accuracy on RTE development set, which is lower than the 86.6% accuracy reported in [124].

	MNLI	RTE	PPDB	Break	avg.
Threeway	**75.6**	**74.4**	**71.2**	85.7	**76.7**
Fourway	**75.6**	74.0	69.8	**86.9**	76.6
Binary (C vs. R)	75.1	72.6	70.5	81.7	75.0
Binary (C/P vs. R)	74.3	72.2	69.8	80.5	74.3

Table 4.4: Comparing binary, threeway, and fourway classification for pretraining.

sibling pair consists of two categories that share the same parent. We also ensure that neutral pairs are non-siblings, meaning that we separate a category that was considered as part of the neutral relations to provide a more fine-grained pretraining signal.

We construct two versions of WikiNLI with binary class labels. One classifies the child against the rest, including parent, neutral, and sibling ("child vs. rest"). The other classifies child or parent against neutral or sibling ("child/parent vs. rest"). The purpose of these two datasets is to find if a more coarse training signal would reduce the gains from pretraining.

These dataset variations are each balanced among their classes and contain 100,000 training instances and 5,000 development instances.

Table 4.4 shows results on the development sets of MNLI, RTE, and PPDB. We report Break results on the test set as it does not have a development set. Overall, fourway and threeway classifications are comparable, although they excel at different tasks. Interestingly, we find that pretraining with child/parent vs. rest is worse than pretraining with child vs. rest. We suspect this is because the child/parent vs. rest task resembles topic classification. The model does not need to determine direction of entailment, but only whether the two phrases are topically related, as neutral pairs are generally either highly unrelated or only vaguely related. The child vs. rest task still requires reasoning about entailment as the models still need to differentiate between child and parent.

	MNLI	RTE	PPDB	Break	avg.
BERT	74.4	71.8	66.9	80.2	73.3
BERT & WikiNLI	**75.6**	**74.4**	**71.2**	85.7	**76.7**
− one cat. layer	74.6	73.3	70.9	**87.0**	76.5
− two cat. layers	75.4	72.9	**71.2**	82.5	75.5
+ page titles	74.2	73.6	70.6	80.7	74.8

Table 4.5: Comparing pruning levels for hierarchies available in Wikipedia.

4.5.2 Wikipedia Pages, Mentions, and Layer Pruning

The variants of WikiNLI we considered so far have used categories as the lowest level of hierarchies. We are interested in whether adding Wikipedia page titles would bring in additional knowledge for inference tasks.

We experiment with including Wikipedia page titles that belong to Wikipedia categories to WikiNLI. We treat these page titles as the leaf nodes of the WikiNLI dataset. Their parents are the categories that the pages belong to.

Although Wikipedia page titles are additional source of information, they are more specific compared to Wikipedia categories. A majority of Wikipedia page titles are person names, locations, or historical events. They are not general summaries of concepts. To explore the effect of more general concepts, we try pruning leaf nodes from the WikiNLI category hierarchies. As higher-level nodes are more general and abstract concepts compared to lower-level nodes, we hypothesize that pruning leaf nodes would make the model learn higher-level concepts. We experiment with pruning one layer and two layers of leaf nodes in WikiNLI category hierarchies.

Table 4.5 compares the results of adding page titles and pruning different numbers of layers. Adding page titles mostly gives relatively small improvements to the model performance on downstream tasks, which shows that the page title is not a useful addition to WikiNLI. Pruning layers also slightly hurts the model performance. One exception is Break, which shows that solving it requires knowledge of higher-level concepts.

Sentence 1	Sentence 2	Rel.
He then moved to **Scottish society** as an actuary for Standard Life Assurance Company. However, he transferred back to London with the company.	He then moved to **Edinburgh** as an actuary for Standard Life Assurance Company. However, he transferred back to London with the company.	parent
Dobroselo () is a village in **Croatia** . It is connected by the D218 highway. According to the 2011 census, Dobroselo had 117 inhabitants.	Dobroselo () is a village in **Southern European countries** . It is connected by the D218 highway. Accordingto the 2011 census, Dobroselo had 117 inhabitants.	child
His oldest brother Charuhasan, like Kamal, is a National Film Award-winning actor who appeared in the **ladino-language** film "Tabarana Kathe".	His oldest brother Charuhasan, like Kamal, is a National Film Award-winning actor who appeared in the **Kannada** film "Tabarana Kathe".	neutral

Table 4.6: Examples from WikiSentNLI.

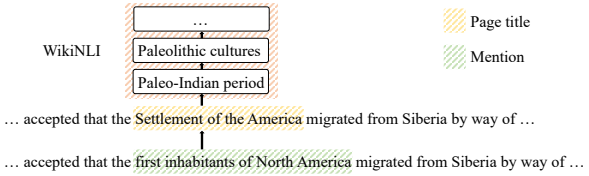


Figure 4.2: An example of WikiSentNLI and higher-level categories that are used to construct WikiNLI.

4.5.3 WikiSentNLI

To investigate the effect of sentential context, we construct another dataset, which we call WikiSentNLI, that is made up of full sentences. The general idea is to create sentence pairs that only differ by several words by using the hyperlinks in the Wikipedia sentences. More specifically, for a sentence with a hyperlink (if there are multiple hyperlinks, we will consider them as different instances), we form new sentences by replacing the text mention (marked by the hyperlink) with the page title as well as the categories describing that page. We consider these two sentences forming candidate child-parent relationship pairs. An example is shown in Figure 4.2. As some page titles or category names do not fit into the context of the sentence, we score them by BERT-Large, averaging over the loss spanning that page title or category name. We pick the candidate with the lowest loss. To generate neutral

	MNLI	RTE	PPDB	Break	avg.
BERT	74.4	71.8	66.9	80.2	73.3
WikiNLI	**76.4**	**74.4**	**71.2**	**85.7**	**76.7**
+ page & mention	72.2	69.0	70.8	58.5	67.6
WikiSentNLI	67.0	62.8	69.1	56.9	64.0
WikiSentNLI cat.	71.8	67.1	70.6	84.0	73.4

Table 4.7: Comparison using WikiSentNLI.

pairs, we randomly sample 20 categories for a particular page mention in the text and pick the candidate with the lowest loss by BERT-Large. WikiSentNLI is also balanced among three relations (child, parent and neutral), and we experiment with 100k training instances and 5k development instances. Table 4.6 are some examples from WikiSentNLI.

Table 4.7 shows the results. In comparing WikiNLI to WikiSentNLI, we observe that adding extra context to WikiNLI does not help on the downstream tasks. It is worth noting that the differences between WikiNLI and WikiSentNLI are more than sentential context. The categories we considered in WikiSentNLI are always immediately after Wikipedia pages, limiting the exposure of higher-level categories.

To look into the importance of those categories, we construct another version of WikiSentNLI by treating the mentions and page title layer as the same level ("WikiSentNLI cat."). This effectively gives models pretrained on this version of WikiSentNLI access to higher-level categories. Practically, when creating child sentences, we randomly choose between keeping the original sentences or replacing the text mention with its linked page title. When creating parent sentences, we replace the text mention with the parent categories of the linked page. Then, we perform the same steps as described in the previous paragraph. Pretraining on WikiSentNLI cat. gives a sizable improvement compared to pretraining on WikiSentNLI.

Additionally, we try to add mentions to WikiNLI, which seems to impair the model performance greatly. This also validates our claim that specific knowledge tends to be noisy and less likely to be helpful for downstream tasks. More interestingly, these variants

	MNLI	RTE	PPDB	Break	avg.
Threeway 100k	75.6	74.4	**71.2**	85.7	76.7
Threeway 400k	**75.7**	**75.5**	70.9	83.0	76.3
Fourway 400k	75.6	75.1	70.8	**89.5**	**77.8**

Table 4.8: The effect of the number of WikiNLI pretraining instances.

	MNLI	RTE	PPDB	Break	avg.
① 100k	75.6	74.4	71.2	85.7	76.7
① 50k + ② 50k	75.0	71.5	70.9	80.2	74.4
① 50k + ③ 50k	75.0	73.6	70.7	81.5	75.3

Table 4.9: Combining WikiNLI with other datasets for pretraining.

seem to affect Break the most, which is in line with our previous finding that Break favors higher-level knowledge. While most of our findings with sentential context are negative, the WikiSentNLI cat. variant shows promising improvements over BERT in some of the downstream tasks, demonstrating that a more appropriate way of incorporating higher-level categories can be essential to benefit from WikiSentNLI in practice.

4.5.4 Larger Training Set

We train on a larger set of WikiNLI dataset, where there are approximately 400,000 training instances, for both threeway and fourway classification settings. We note that we only pretrain models on WikiNLI for one epoch as it leads to better performance on downstream tasks. The results are in Table 4.8. We observe that except for PPDB, adding more data generally improves performance. For Break, we observe significant improvements when using fourway WikiNLI for pretraining, whereas threeway WikiNLI seems to hurt the performance.

4.5.5 Combining Multiple Data Sources

We combine multiple data sources for pretraining. In one setting we combine 50k instances of WikiNLI with 50k instances of WordNet, while in the other setting we combine 50k instances

phrase 1	phrase 2	gold	BERT	WikiNLI	WordNet	Wikidata
car	the trunk	hypernym	other	hypernym	hypernym	hypernym
return	return home	hypernym	synonym	hypernym	hypernym	hypernym
boys are	the children are	hyponym	synonym	hyponym	hyponym	hyponym
foreign affairs	foreign minister	other	hypernym	other	hypernym	hypernym
company	debt	other	independent	independent	other	other
europe	japan	alternation	hypernym	alternation	independent	alternation
family	woman	independent	independent	hypernym	independent	other

Table 4.10: Examples from PPDB development set showing the effect of pretraining resources. "other" stands for "other-related"

	2000	3000	5000	10000	20000
BERT	72.2	74.4	76.6	78.8	80.4
WikiNLI	74.5	75.6	77.3	79.1	80.6
	+2.3	+1.2	+0.7	+0.3	+0.2

Table 4.11: Results for varying numbers of MNLI training instances.

of WikiNLI with 50k instances of Wikidata. Table 4.9 compares these two settings for pretraining. In this table ①=WikiNLI;②=WordNet;③=Wikidata. WikiNLI works the best when pretraining alone.

4.5.6 Effect of Pretraining Resources

We show several examples of predictions from PPDB in Table 4.10. In general, we observe that without pretraining, BERT tends to predict symmetric categories, such as synonym, or other-related, instead of predicting entailment-related categories. For example, the phrase pair "car" and "the trunk", "return" and "return home", and "boys are" and "the children are". These are either "hypernym" or "hyponym" relationship, but BERT tends to conflate them with symmetric relationships, such as other-related. To quantify this hypothesis, we compute the numbers of correctly predicted antonym, alternation, hyponym and hypernym and show them in Table 4.12. It can be seen that with pretraining those numbers increase dramatically, showing the benefit of pretraining on these resources.

	antonym	alternation	hyponym	hypernym
w/	34	51	276	346
w/o	1	35	231	248

Table 4.12: Per category numbers of correctly predicted instances by BERT with or without pretraining on WikiNLI.

We also observe that the model performance can be affected by the coverage of pretraining resources. In particular, for phrase pair "foreign affairs" and "foreign minister", WikiNLI has a closely related term "foreign affair ministries" and "foreign minister" under the category "international relations", whereas WordNet does not have these two, and Wikidata only has "foreign minister".

In addition, for phrase pair "company" and "debt", in WikiNLI, the company is under the "business" category; debt is under the "finance" category. They are not directly related, whereas in WordNet, due to the multisense of company, company and debt are both treated as a kind of "state", and in Wikidata, they are both a subclass of "legal concept".

For phrase pair "family" and "woman", in WikiNLI, "family" is a parent category of "wives", and in Wikidata, they are related in that the "family" is a subclass of "group of humans". In contrast, WordNet does not have such knowledge.

4.5.7 Finetuning with More Data

Pretraining on WikiNLI has more significant improvement with less training data. Table 4.11 shows the model improvement to BERT-Large with WikiNLI when training on 2000, 3000, 5000, 10000, and 20000 MNLI training instances accordingly. The gap between BERT-Large and WikiNLI narrows as the MNLI training data size increases.

4.6 Summary

In this chapter, we described WikiNLI, a large-scale naturally-annotated dataset for improving model performance on NLI and LE tasks. Empirically, we benchmarked WordNet, Wikidata, and WikiNLI using both BERT and RoBERTa by first pretraining these models on those resources, then finetuning on downstream tasks. The results showed that pretraining on WikiNLI gives the largest gains averaging over 10 different datasets. The improvements to both BERT and RoBERTa showed that the benefit of WikiNLI can generalize. We also performed an in-depth analysis on ways of handling the Wikipedia category graph, including pruning lower-level categories, adding sentential context and pretraining with more instances.

CHAPTER 5

LEARNING ENTITY REPRESENTATIONS

This chapter describes learning entity representations from Wikipedia documents and their inter-document hyperlink structures.

5.1 Introduction

Entity representations play a key role in numerous important problems including language modeling [87], dialogue generation [69], entity linking [67], and story generation [39]. One successful line of work on learning entity representations has been learning *static* embeddings: that is, assign a unique vector to each entity in the training data [67, 230, 231]. While these embeddings are useful in many applications, they have the obvious drawback of not accommodating unknown entities. Another limiting factor is the lack of an evaluation benchmark: it is often difficult to know which entity representations are better for which tasks.

In this chapter, we introduce EntEval: a carefully designed benchmark for holistically evaluating entity representations. It is a test suite of diverse tasks that require nontrivial understanding of entities, including entity typing, entity similarity, entity relation prediction, and entity disambiguation. Motivated by the recent success of contextualized word representations (henceforth: CWRs) from pretrained models [140, 171, 47, 232, 124], we propose to encode the mention context or the description to dynamically represent an entity. In addition, we perform an in-depth comparison of ELMo and BERT-based embeddings and find that they show different characteristics on different tasks. We analyze each layer of the CWRs and make the following observations:

- The dynamically encoded entity representations show a strong improvement on the entity disambiguation task compared to prior work using static entity embeddings.

- BERT-based entity representations require further supervised training to perform well on downstream tasks, while ELMo-based representations are more capable of performing zero-shot tasks.
- In general, higher layers of ELMo and BERT-based CWRs are more transferable to EntEval tasks.

To further improve contextualized and descriptive entity representations (CER/DER), we leverage natural hyperlink annotations in Wikipedia. We identify effective objectives for incorporating the contextual information in hyperlinks and improve ELMo-based CWRs on a variety of entity related tasks.

5.2 EntEval

We are interested in two approaches: contextualized entity representations (henceforth: CER) and descriptive entity representations (henceforth: DER), both encoding fixed-length vector representations for entities.

The contextualized entity representations encodes an entity based on the context it appears regardless of whether the entity is seen before. The motivation behind contextualized entity representations is that we want an entity encoder that does not depend on entries in a knowledge base, but is capable of inferring knowledge about an entity from the context it appears.

As opposed to contextualized entity representations, descriptive entity representations do rely on entries in Wikipedia. We use a model-specific function f to obtain a fixed-length vector representation from the entity's textual description.

To evaluate CERs and DERs, we propose a wide range of entity related tasks. Since our purpose is for examining the learned entity representations, we only use a linear classifier and freeze the entity representations when performing the following tasks. Unless otherwise noted, when the task involves a pair of entities, the input to the classifier are the entity repre-

	CAP		CERP	EFP	ET	KORE	WikiSRS		ERT	Rare	CoNLL
	same	next					Rel	Sim			
#train	3982	3982	4000	10000	1998	N/A	N/A	N/A	3130	10000	18538
#valid	3806	3828	4000	2000	1998	N/A	N/A	N/A	6260	4000	4790
#test	3938	3850	4000	2000	1998	20×20	688	688	6260	4000	4481
#classes	2		2	2	10331	N/A	N/A	N/A	626	4	up to 30

Table 5.1: Statistics of datasets used in EntEval tasks. CAP: coreference arc prediction, CERP: contexualized entity relationship prediction, EFP: entity factuality prediction, ET: entity typing, ESR: entity similarity and relatedness, ERT: entity relationship typing, NED: named entity disambiguation, Rare: rare entity prediction, CoNLL: CoNLL-YAGO named entity disambiguation.

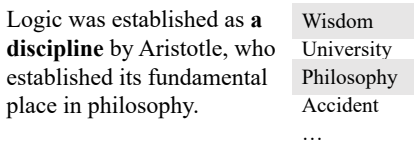


Figure 5.1: An example taken from ET. Targeted entity mention is bold. Candidate categories are on the right. Gold standard categories are in gray.

sentations x_1 and x_2, concatenated with their element-wise product and absolute difference: $[x_1, x_2, x_1 \odot x_2, |x_1 - x_2|]$. This input format has been used in SentEval [40].

The datasets used in EntEval tasks are summarized in table 5.1. It shows the number of instances in train/valid/test split for each dataset, and the number of target classes if this is a classification task. We describe the proposed tasks in the following subsections.

5.2.1 Entity Typing (ET)

The task of entity typing (ET) is to assign types to an entity given only the context of the entity mention. ET is context-sensitive, making it an effective approach to probe the knowledge of context encoded in pretrained representations. For example, in the sentence "Bill Gates has donated billions to eradicate malaria", "Bill Gates" has the type of "philanthropist" instead of "inventor" [36].

In this task, we will contextualized entity representations, followed by a linear layer to make predictions. We use the annotated ultra-fine entity typing dataset of [36] with standard

data splits. As shown in Figure 5.1, there can be multiple labels for an instance. We use binary log loss for training using all positive and negative entity types, and report F_1 score. Thresholds are tuned based on validation set accuracy.

5.2.2 Coreference Arc Prediction (CAP)

Given two entities and the associated context, the task is to determine whether they refer to the same entity. Solving this task may require the knowledge of entities. For example, in the sentence "Revenues of \$14.5 billion were posted by Dell_1. The company_1 ...", there is no prior context of "Dell", so having known "Dell" is a company instead of the people "Michael Dell" will surely benefit the model [51]. Unlike other tasks, coreference typically involves longer context. To restrict the effect of broad context, we only keep two groups of coreference arcs from smaller context. One includes mentions that are in the same sentence ("same") for examining the model capability of encoding local context. The other includes mentions that are in consecutive sentences ("next") for the broader context. We create this task from the PreCo dataset [26], which has mentions annotated even when they are not part of coreference chains. We filter out instances in which both mentions are pronouns. All non-coreferent mention pairs are considered to be negative samples.

To make this task more challenging, for each instance we compute cosine similarity of mentions by averaging GloVe word vectors. We group the instances into bins by cosine similarity, and randomly select the same number of positive and negative instances from each bin to ensure that models do not solve this task by simply comparing similarity of mention names.

We use the contextualized entity representations of the two mentions to infer coreference arcs with supervised training and report the averaged accuracy of "same" and "next".

REFUTES: The **New York City Landmarks Preservation Commission** consists of zero commissioners.
SUPPORTS: **TD Garden** has held Bruins games.

Figure 5.2: Two examples from the EFP.

TRUE: **Gin and vermouth** can **make a martini**
FALSE: **Connecticut** is not **a state**

Figure 5.3: Examples from the CERP.

5.2.3 Entity Factuality Prediction (EFP)

The entity factuality prediction (EFP) task involves determining the correctness of statements regarding entities. We use the manually-annotated FEVER dataset [206] for this task. FEVER is a task to verify whether a statement is supported by evidences. The original FEVER dataset includes three classes, namely "Supports", "Refutes", and "NotEnoughInfo" and evidences are additionally available for each instance. As our purpose is to examine the knowledge encoded in entity representations, we discard the last category ("NotEnoughInfo") and the evidence. In rare cases, instances in FEVER may include multiple entity mentions, so we randomly pick one. We randomly sample 10000, 2000, and 2000 instances for our training, validation, and test sets, respectively.

In this task, entity representations can be obtained either by contextualized entity representations or descriptive entity representations. In practice, we observe descriptive entity representations give better performance, which presumably is because these statements are more similar to descriptions than entity mentions. As shown in Figure 5.2, without providing additional evidences, solving this task requires knowledge of entities encoded in representations. We directly use entity representations as input to the classifier.

5.2.4 Contexualized Entity Relationship Prediction (CERP)

The task of contexualized entity relationship prediction (CERP) modeling determines the connection between two entities appeared in the same context.

We use sentences from ConceptNet [197] with automatically parsed mentions and templates used to construct the dataset. We filter out non-English concepts and relations such as 'related', 'translation', 'synonym', and 'likely to find' since we seek to evaluate more complicated knowledge of entities encoded in representations. We further filter out non-entity mentions and entities with type 'DATE', 'TIME', 'PERCENT', 'MONEY', 'QUANTITY', 'ORDINAL', and 'CARDINAL' according to SpaCy [78]. After filtering, we have 13374 assertions.

Negative samples are generated based on the following rules:

1. For each relationship, we replace an entity with similar negative entities based on cosine similarity of averaged GloVe embeddings [167].
2. We change the relationship in positive samples from affirmation to negation (e.g., 'is' to 'is not'). These serve as negative samples.
3. We further sample positive samples from (1) in an attempt to prevent the 'not' token from being biased towards negative samples. Therefore, for negative samples we get from (1), we change the relationship from affirmation to negation as in (2) to get positive samples.

For example, let 'A is B' be the positive sample. (1) changes it to 'C is B' which serves as a negative sample and (2) changes it to 'A is not B' as another negative sample. (3) changes it to 'C is not B' as a positive example. In the end, we randomly sample 6000 instances from each class. This ends up yielding a 4000/4000/4000 train/dev/test dataset. As shown in Figure 5.3, this task cannot be solved by relying on surface form of sentences, instead it requires the input representations to encode knowledge of entities based on the context.

We use contextualized entity representations in this task.

Score	Entity Name
-	Apple Inc.
20	Steve Jobs
...	...
11	Microsoft
...	...
1	Ford Motor Company

Table 5.2: An example from KORE.

5.2.5 Entity Similarity and Relatedness (ESR)

Given two entities with their descriptions from Wikipedia, the task is to determine their similarity or relatedness. After the entity descriptions are encoded into vector representations, we compute their cosine similarity as predictions. We use the KORE [76] and WikiSRS [153] datasets in this task. Since the original datasets only provide entity names, we automatically add Wikipedia descriptions to each entity and manually ensure that every entity is matched to a Wikipedia description. We use Spearman's rank correlation coefficient between our computed cosine similarity and the gold standard similarity/relatedness scores to measure the performance of entity representations.

The task of KORE is to rank the candidate entities by similarity. As KORE does not provide similarity scores of entity pairs, but simply ranks the candidate entities by their similarities to a target entity, we assign scores from 20 to 1 accordingly to each entity in the order of similarity. Table 5.2 shows an example from KORE. The fact that "Apple Inc." is more related to "Steve Jobs" than "Microsoft" requires multiple steps of inference, which motivates this task. Since the predictor we use is cosine similarity, which does not introduce additional parameters, we directly use encoded representations on the test set without any supervised training.

Figure 5.4: An example from CoNLL-YAGO.

5.2.6 Entity Relationship Typing (ERT)

As another popular resource for common knowledge, we consider using Freebase [18] for probing the encoded knowledge by classifying the types of relations between pair of entities. First, we extract entity relation tuples (entity1, relation, entity2) from Freebase and then filter out easy tuples based on training a classifier using averaged GloVe vectors of entity names as input, which leaves us 626 types of relations, including "internet.website.owner", "film.film_art_director.films_art_directed", and "comic_books.comic_book_series.genre". We randomly sample 5 instances for each relation type to form our training set and 10 instances per type the for validation and test sets. We use Wikipedia descriptions for each entity in the pair whose relation we are predicting and we use descriptive entity representations for each entity with supervised training.

5.2.7 Named Entity Disambiguation (NED)

Named entity disambiguation is the task of linking a named-entity mention to its corresponding instance in a knowledge base such as Wikipedia. In this task, we consider CoNLL-YAGO (CoNLL; 77) and Rare Entity Prediction (Rare; 130).

For CoNLL-YAGO, following [77] and [230], we used the 27,816 mentions with valid entries in the knowledge base. For each entity mention m in its context, we generate a set of (at most) its top 30 candidate entities $C_m = \{c_j\}$ using CrossWikis [198]. Some gold standard candidates c are not present in CrossWikis, so we set the prior probability $p_{\text{prior}}(y)$ for those to 1e-6 and normalize the resulting priors for the candidate entities. When adding

Wikipedia descriptions, we manually ensure gold standard mentions are attached to a description, however, we discard candidate mentions that cannot be aligned to a Wikipedia page. We use contextualized entity representations for entity mentions and use descriptive entity representations for candidate entities. Training minimizes binary log loss using all negative examples. At test time, we use $\arg\max_{c \in C_m}[p_{\text{prior}}(c) + p_{\text{classifier}}(c)]$ as the prediction. We note that directly using prior as predictions yields an accuracy of 58.2%.

[130] introduce the task of *rare entity prediction*. The task has a similar format to CoNLL-YAGO entity linking. Given a document with a blank in it, the task is to select an entity from a provided list of entities with descriptions. Only rare entities are used in this dataset so that performing well on the task requires the ability to effectively represent entity descriptions. We randomly select 10k/4k/4k examples to construct train/valid/test sets. For simplicity, we only keep instances with four candidate entities.

Figure 5.4 shows an example from CoNLL-YAGO, where the "China" in context has many deceptive meanings. Here the candidate "China" has exact string match of the entity name but it should not be selected as it is an after-game report on soccer. To match the entities, this task requires both effective contextualize entity representations and descriptive entity representation.

Practically, we encode the context using CER to be x_1, and encode each entity description using DER to be x_2, and pass $[x_1, x_2, x_1 \odot x_2, |x_1 - x_2|]$ to a linear model to predict whether it is the correct entity to fill in. The model is trained with cross entropy loss.

5.3 Methods

We first describe how we define encoders for contextualized entity representations (Section 5.3.1) and descriptive entity representations (Section 5.3.2), then we discuss how we train new encoders tailored to capture information from the hyperlink structure of Wikipedia (Section 5.3.3).

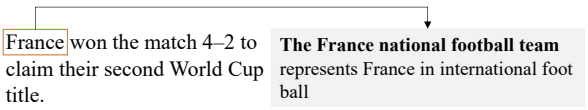


Figure 5.5: An example of hyperlinks in Wikipedia.

5.3.1 Encoders for Contextualized Entity Representations

For defining these encoders, we assume we have a sentence $s = (w_1, \ldots, w_T)$ where span $(w_i, \ldots, w_j)$ refers to an entity mention. When using ELMo, we first encode the sentence: $(c_1, \ldots, c_T) = \text{ELMo}(w_1, \cdots, w_T)$, and we use the average of contextualized hidden states corresponding to the entity span as the contextualized entity representation. That is, $f_{\text{ELMo}}(w_{1:T}, i, j) = \frac{\sum_{k=i}^{j} c_k}{j-i+1}$.

With BERT, following [158], we concatenate the full sentence with the entity mention, starting with [CLS] and separating by [SEP], i.e., $[\text{CLS}], w_1, \ldots, w_T, [\text{SEP}], w_i, \ldots, w_j, [\text{SEP}]$. We encode the full sequence using BERT and use the output from the [CLS] token as the entity mention representation.

5.3.2 Encoders for Descriptive Entity Representations

We encode an entity description by treating the entity description as a sentence, and use the average of the hidden states from ELMo as the entity description representation. With BERT, we use the output from the [CLS] token as the description representation.

5.3.3 Hyperlink-Based Training

An entity mentioned in a Wikipedia article is often linked to its Wikipedia page, which provides a useful description of the mentioned entity. The same Wikipedia page may correspond to many different entity mentions. Likewise, the same entity mention may refer to different Wikipedia pages depending on its context. For instance, as shown in Figure 5.5, based on the context, "France" is linked to the Wikipedia page of "France national football team"

instead of the country. The specific entity in the knowledge base can be inferred from the context information. In such cases, we believe Wikipedia provides valuable complementary information to the current pretrained CWRs such as BERT and ELMo.

To incorporate such information during training, we automatically construct a hyperlink-enriched dataset from Wikipedia that we will refer to as WikiEnt. Prior work has used similar resources [192, 67], but we aim to standardize the process and will release the dataset.

The WikiEnt dataset consists of sentences with contextualized entity mentions and their corresponding descriptions obtained via hyperlinked Wikipedia pages. When processing descriptions, we only keep the first 100 word tokens at most as the description of a Wikipedia page; similar truncation has been done in prior work [67]. For context sentences, we remove those without hyperlinks from the training data and duplicate those with multiple hyperlinks. We also remove context sentences for which we cannot find matched Wikipedia descriptions. These processing steps result in a training set of approximately 85 million instances and over 3 million unique entities.

We define a hyperlink-based training objective and add it to ELMo. In particular, we use contextualized entity representations to decode the hyperlinked Wikipedia description, and also use the descriptive entity representations to decode the linked context. We use bag-of-words decoders in both decoding processes. More specifically, given a context sentence $x_{1:T_x}$ with mention span (i, j) and a description sentence $y_{1:T_y}$, we use the same bidirectional language modeling loss $l_{\text{lang}}(x_{1:T_x}) + l_{\text{lang}}(y_{1:T_y})$ in ELMo where

$$l_{\text{lang}}(u_{1:T}) = -\sum_{t=1}^{T} \log p(u_{t+1}|u_1, \ldots, u_t) + \log p(u_{t-1}|u_t, \ldots, u_T)$$

and p is defined by the ELMo parameters. In addition, we define the two bag-of-words

	CAP	CERP	EFP	ET	ESR	ERT	NED	Average
GloVe	71.9	52.6	67.0	10.3	50.9	40.8	41.2	47.8
BERT Base	**80.6**	65.6	74.8	32.0	28.8	42.2	50.6	53.5
BERT Large	79.1	**66.9**	**76.7**	32.3	32.6	**48.8**	**54.3**	55.8
ELMo	80.2	61.2	75.8	**35.6**	60.3	46.8	51.6	**58.8**
EntELMo baseline	78.0	59.6	71.5	31.3	**61.6**	46.5	48.5	56.7
EntELMo	76.9	59.9	72.4	32.2	59.7	45.7	49.0	56.5
EntELMo w/o l_{ctx}	73.5	59.4	71.1	33.2	53.3	44.6	48.9	54.9
EntELMo w/ l_{etn}	76.2	60.4	70.9	33.6	49.0	42.9	49.3	54.6

Table 5.3: Performances of entity representations on EntEval tasks.

reconstruction losses:

$$l_{\text{ctx}} = -\sum_t \log q(x_t | f_{\text{ELMo}}([\text{BOD}] y_{1:T_y}, 1, T_y))$$

$$l_{\text{desc}} = -\sum_t \log q(y_t | f_{\text{ELMo}}([\text{BOC}] x_{1:T_x}, i, j))$$

where [BOD] and [BOC] are special symbols prepended to sentences to distinguish descriptions from contexts. The distribution q is parameterized by a linear layer that transforms the conditioning embedding into weights over the vocabulary. The final training loss is

$$l_{\text{lang}}(x_{1:T_x}) + l_{\text{lang}}(y_{1:T_y}) + l_{\text{ctx}} + l_{\text{desc}} \tag{5.1}$$

Same as the original ELMo, each log loss is approximated with negative sampling [82]. We write EntELMo to denote the model trained by Eq. (5.1). When using EntELMo for contextualized entity representations and descriptive entity representations, we use it analogously to ELMo.

5.4 Experiments

5.4.1 Setup

As a baseline for hyperlink-based training, we train EntELMo on the WikiEnt dataset with only a bidirectional language model loss. Due to the limitation of computational resources, both variants of EntELMo are trained for one epoch (3 weeks time) with smaller dimensions than ELMo. We set the hidden dimension of each directional long short-term memory network (LSTM; 74) layer to be 600, and project it to 300 dimensions. The resulting vectors from each layer are thus 600 dimensional. We use 1024 as the negative sampling size for each positive word token. For bag-of-words reconstruction, we randomly sample at most 50 word tokens as positive samples from the the target word tokens. Other hyperparameters are the same as ELMo. EntELMo is implemented based on the official ELMo implementation.[1]

As a baseline for contextualized and descriptive entity representations, we use GloVe word averaging of the entity mention as the "contextualized" entity representation, and use word averaging of the truncated entity description text as its description representation. We also experiment two variants of EntELMo, namely EntELMo w/o l_{ctx} and EntELMo with l_{etn}. For second variant, we replace l_{ctx} with l_{etn}, where we only decode entity mentions instead of the whole context from descriptions. We lowercased all training data as well as the evaluation benchmarks.

We evaluate the transferrability of ELMo, EntELMo, and BERT by using trainable mixing weights for each layer. For ELMo and EntELMo, we follow the recommendation from [171] to first pass mixing weights through a softmax layer and then multiply the weighted-summed representations by a scalar. For BERT, we find it better to just use unnormalized mixing weights. In addition, we investigate per-layer performance for both models in Section 5.5.

1. Our implementation is available at `https://github.com/mingdachen/bilm-tf`

5.4.2 Results

Table 5.3 shows the performance of our models on the EntEval tasks. Our findings are detailed below:

- Pretrained CWRs (ELMo, BERT) perform the best on EntEval overall, indicating that they capture knowledge about entities in contextual mentions or as entity descriptions.
- BERT performs poorly on entity similarity and relatedness tasks. Since this task is zero-shot, it validates the recommended setting of finetuning BERT [47] on downstream tasks, while the embedding of the [CLS] token does not necessarily capture the semantics of the entity.
- BERT Large is better than BERT Base on average, showing large improvements in ERT and NED. To perform well at ERT, a model must either glean particular relationships from pairs of lengthy entity descriptions or else leverage knowledge from pretraining about the entities considered. Relatedly, performance on NED is expected to increase with both the ability to extract knowledge from descriptions and by starting with increased knowledge from pretraining. The Large model appears to be handling these capabilities better than the Base model.
- EntELMo improves over the EntELMo baseline (trained without the hyperlinking loss) on some tasks but suffers on others. The hyperlink-based training helps on CERP, EFP, ET, and NED. Since the hyperlink loss is closely-associated to the NED problem, it is unsurprising that NED performance is improved. Overall, we believe that hyperlink-based training benefits contextualized entity representations but does not benefit descriptive entity representations (see, for example, the drop of nearly 2 points on ESR, which is based solely on descriptive representations). This pattern may be due to the difficulty of using descriptive entity representations to reconstruct their appearing context.

	Rare		CoNLL		ERT	
	Des.	Name	Des.	Name	Des.	Name
ELMo	38.1	36.7	63.4	71.2	46.8	31.5
BERT Base	42.2	36.6	64.7	74.3	42.2	34.3
BERT Large	48.8	44.0	64.6	74.8	48.8	32.6

Table 5.4: Accuracies (%) in comparing the use of description encoder (Des.) to entity name (Name).

	CoNLL
ELMo	71.2
[67]	65.1
Deep ED	66.7

Table 5.5: Accuracies (%) on CoNLL-YAGO with static or non-static entity representations.

5.5 Analysis

Is descriptive entity representation necessary? A natural question to ask is whether the entity description is needed, as for humans, the entity names carry sufficient amount of information for a lot of tasks. To answer this question, we experiment with encoding entity names by the descriptive entity encoder for ERT (entity relationship typing) and NED (named entity disambiguation) tasks. The results in Table 5.4 show that encoding the entity names by themselves already captures a great deal of knowledge regarding entities, especially for CoNLL-YAGO. However, in tasks like ERT, the entity descriptions are crucial as the names do not reveal enough information to categorize their relationships.

Table 5.5 reports the performance of different descriptive entity representations on the CoNLL-YAGO task. The three models all use ELMo as the context encoder. "ELMo" encodes the entity name with ELMo as descriptive encoder, while both [67] and Deep ED [61] use their trained static entity embeddings. [2] As [67] and Deep ED have different embedding sizes from ELMo, we add an extra linear layer after them to map to the same dimension.

2. We note that the numbers reported here are not strictly comparable to the ones in their original paper since we keep all the top 30 candidates from Crosswiki while prior work employs different pruning heuristics.

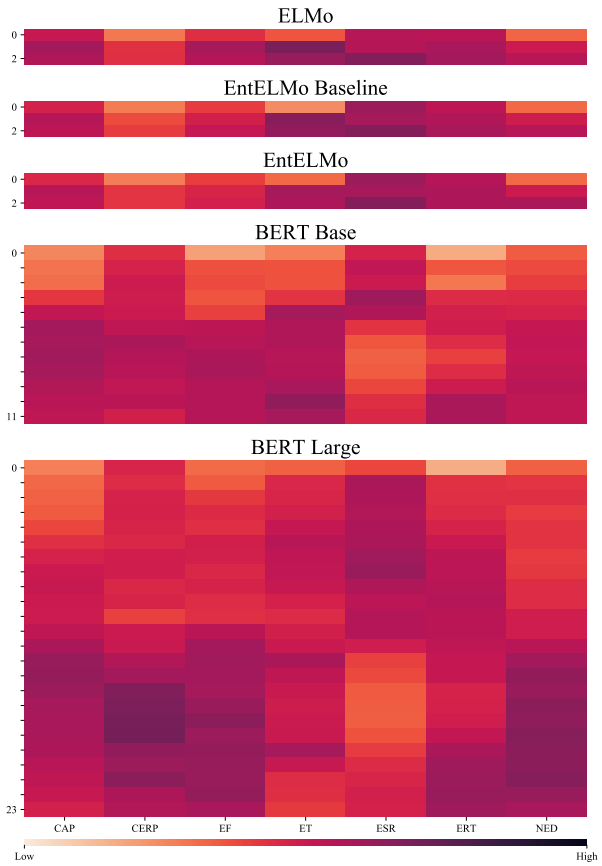


Figure 5.6: Heatmap showing per-layer performances for ELMo, EntELMo baseline, EntELMo, BERT Base, and BERT Large.

These two models are designed for entity linking, which gives them potential advantages. Even so, ELMo outperforms them both by a wide margin.

Per-Layer Analysis. We evaluate each ELMo and EntELMo layer, i.e., the character CNN layer and two bidirectional LSTM layers, as well as each BERT layer on the EntEval tasks. Figure 5.6 reveals that for ELMo models, the first and second LSTM layers capture most of the entity knowledge from context and descriptions. The BERT layers show more diversity. Lower layers perform better on ESR (entity similarity and relatedness), while for other tasks higher layers are more effective.

5.6 Summary

Our proposed EntEval test suite provides a standardized evaluation method for entity representations. We demonstrate that EntEval tasks can benefit from the success of contextualized word representations such as ELMo and BERT. Augmenting encoding-decoding loss leveraging natural hyperlinks from Wikipedia further improves ELMo on some EntEval tasks. As shown by our experimental results, the contextualized entity encoder benefits more from this hyperlink-based training objective, suggesting future works to prioritize encoding entity description from its mention context.

CHAPTER 6

LEARNING DISCOURSE SENTENCE REPRESENTATIONS

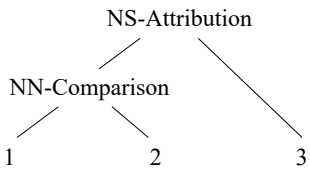
In this chapter, we seek to incorporate and evaluate discourse knowledge in general purpose sentence representations. We also propose DiscoEval, a task suite designed to evaluate discourse-related knowledge in pretrained sentence representations.

6.1 Introduction

A discourse is a coherent, structured group of sentences that acts as a fundamental type of structure in natural language [93]. A discourse structure is often characterized by the arrangement of semantic elements across multiple sentences, such as entities and pronouns. The simplest such arrangement (i.e., linearly-structured) can be understood as sentence ordering, where the structure is manifested in the timing of introducing entities. Deeper discourse structures use more complex relations among sentences (e.g., tree-structured; see Figure 6.1).

Theoretically, discourse structures have been approached through Centering Theory [65] for studying distributions of entities across text and Rhetorical Structure Theory (RST; 134) for modelling the logical structure of natural language via discourse trees. Researchers have found modelling discourse useful in a range of tasks [68, 151, 123, 159], including summarization [62], text classification [86], and text generation [19].

In this chapter, we describe DiscoEval, a task suite designed to evaluate discourse-related knowledge in pretrained sentence representations. DiscoEval comprises 7 task groups covering multiple domains, including Wikipedia, stories, dialogues, and scientific literature. The tasks are probing tasks [188, 2, 14, 172, 42, 176, 205, 122, 54, 30, *inter alia*] based on sentence ordering, annotated discourse relations, and discourse coherence. The data is either generated semi-automatically or based on human annotations [23, 177, 118, 101].



[The European Community's consumer price index rose a provisional 0.6% in September from August]$_1$ [and was up 5.3% from September 1988,]$_2$ [according to Eurostat, the EC's statistical agency.]$_3$

Figure 6.1: An RST discourse tree from the RST Discourse Treebank. "N" represents "nucleus", containing basic information for the relation. "S" represents "satellite", containing additional information about the nucleus.

We also propose a set of novel multi-task learning objectives building upon standard pretrained sentence encoders, which rely on the assumption of distributional semantics of text. These objectives depend only on the natural structure in structured document collections like Wikipedia.

Empirically, we benchmark our models and several popular sentence encoders on DiscoEval and SentEval [40]. We find that our proposed training objectives help the models capture different characteristics in the sentence representations. Additionally, we find that ELMo shows strong performance on SentEval, whereas BERT performs the best among the pretrained embeddings on DiscoEval. Both BERT and Skip-thought vectors [98], which have training losses explicitly related to surrounding sentences, perform much stronger compared to their respective prior work, demonstrating the effectiveness of incorporating losses that make use of broader context. Through per-layer analysis, we also find that for both BERT and ELMo, deep layers consistently outperform shallower ones on DiscoEval, showing different trends from SentEval where the shallow layers have the best performance.

6.2 Discourse Evaluation

We propose DiscoEval, a test suite of 7 tasks to evaluate whether sentence representations include semantic information relevant to discourse processing. Below we describe the tasks and datasets, as well as the evaluation framework. We closely follow the SentEval sentence

embedding evaluation suite, in particular its supervised sentence and sentence pair classification tasks, which use predefined neural architectures with slots for fixed-dimensional sentence embeddings. All DiscoEval tasks are modelled by logistic regression unless otherwise stated in later sections.

We also experimented with adding hidden layers to the DiscoEval classification models. However, we find simpler linear classifiers to provide a clearer comparison among sentence embedding methods. More complex classification models lead to noisier results, as more of the modelling burden is shifted to the optimization of the classifiers. Hence we decide to evaluate the sentence embeddings with simple classification models.

In the rest of this section, we will use $[\cdot, \cdot, \cdots]$ to denote concatenation of vectors, $\odot$ for element-wise multiplication, and $|\cdot|$ for element-wise absolute value.

6.2.1 Discourse Relations

As the most direct way to probe discourse knowledge, we consider the task of predicting annotated discourse relations among sentences. We use two human-annotated datasets: the RST Discourse Treebank (RST-DT; 23) and the Penn Discourse Treebank (PDTB; 177). They have different labeling schemes. PDTB provides discourse markers for adjacent sentences, whereas RST-DT offers document-level discourse trees, which recently was used to evaluate discourse knowledge encoded in document-level models [57]. The difference allows us to see if the pretrained representations capture local or global information about discourse structure.

More specifically, as shown in Figure 6.1, in RST-DT, text is segmented into basic units, elementary discourse units (EDUs), upon which a discourse tree is built recursively. Although a relation can take multiple units, we follow prior work [84] to use right-branching trees for non-binary relations to binarize the tree structure and use the 18 coarse-grained relations defined by [23].

1. In any case, the brokerage firms are clearly moving faster to create new ads than they did in the fall of 1987.
2. [But] it remains to be seen whether their ads will be any more effective.
label: Comparison.Contrast

Figure 6.2: Example in the PDTB explicit relation task.

When evaluating pretrained sentence encoders on RST-DT, we first encode EDUs into vectors, then use averaged vectors of EDUs of subtrees as the representation of the subtrees. The target prediction is the label of nodes in discourse trees and the input to the classifier is $[x_{\text{left}}, x_{\text{right}}, x_{\text{left}} \odot x_{\text{right}}, |x_{\text{left}} - x_{\text{right}}|]$, where x_{left} and x_{right} are vector representations of the left and right subtrees respectively. For example, the input for target "NN-Attribution" in Figure 6.1 would be $x_{\text{left}} = \frac{x_1+x_2}{2}$, $x_{\text{right}} = x_3$, where x_i is the encoded representation for the ith EDU in the text. We use the standard data splits, where there are 347 documents for training and 38 documents for testing. We choose 35 documents from the training set to serve as a validation set.

For PDTB, we use a pair of sentences to predict discourse relations. Following [118], we focus on two kinds of relations from PDTB: explicit (PDTB-E) and implicit (PDTB-I). The sentence pairs with explicit relations are two consecutive sentences with a particular connective word in between. Figure 6.2 is an example of an explicit relation. The words in [] are taken out from input sentence 2.

In the PDTB, annotators insert an implicit connective between adjacent sentences to reflect their relations, if such an implicit relation exists. Figure 6.3 shows an example of an implicit relation. The PDTB provides a three-level hierarchy of relation tags. In DiscoEval, we use the second level of types [118], as they provide finer semantic distinctions compared to the first level. To ensure there is a reasonable amount of evaluation data, we use sections 2-14 as training set, 15-18 as development set, and 19-23 as test set. In addition, we filter out categories that have less than 10 instances. This leaves us 12 categories for explicit relations and 11 for implicit ones. Category names are listed in the supplementary material.

1. “A lot of investor confidence comes from the fact that they can speak to us,” he says.
2. [so] “To maintain that dialogue is absolutely crucial.”
label: Contingency.Cause

Figure 6.3: Example in the PDTB implicit relation task.

We use the sentence embeddings to infer sentence relations with supervised training. As input to the classifier, we encode both sentences to vector representations x_1 and x_2, concatenated with their element-wise product and absolute difference: $[x_1, x_2, x_1 \odot x_2, |x_1 - x_2|]$.

6.2.2 Sentence Position (SP)

We create a task that we call Sentence Position. It can be seen as way to probe the knowledge of linearly-structured discourse, where the ordering corresponds to the timings of events. When constructing this dataset, we take five consecutive sentences from a corpus, randomly move one of these five sentences to the first position, and ask models to predict the true position of the first sentence in the modified sequence.

We create three versions of this task, one for each of the following three domains: the first five sentences of the introduction section of a Wikipedia article (Wiki), the ROC Stories corpus (ROC; 147), and the first 5 sentences in the abstracts of arXiv papers (arXiv; 33). Figure 6.4 shows an example of this task for the ROC Stories domain. The first sentence should be in the fourth position among these sentences. To make correct predictions, the model needs to be aware of both typical orderings of events as well as how events are described in language. In the example shown, Bonnie's excitement comes from her imagination so it must happen after she picked up the jeans and tried them on but right before she realized the actual size.

To train classifiers for these tasks, we do the following. We first encode the five sentences to vector representations x_i. As input to the classifier, we include x_1 and the concatenation

- She was excited thinking she must have lost weight. - Bonnie hated trying on clothes. - She picked up a pair of size 12 jeans from the display. - When she tried them on they were too big! - Then she realized they actually size 14s, and 12s.

Figure 6.4: Example from the ROC Stories domain of the Sentence Position task.

1. These functions include fast and synchronized response to environmental change, or long-term memory about the transcriptional status. 2. Focusing on the collective behaviors on a population level, we explore potential regulatory functions this model can offer.

Figure 6.5: Example from the arXiv domain of the Binary Sentence Ordering task (incorrect ordering shown).

of $x_1 - x_i$ for all i: $[x_1, x_1 - x_2, x_1 - x_3, x_1 - x_4, x_1 - x_5]$.

6.2.3 Binary Sentence Ordering (BSO)

Similar to sentence position prediction, Binary Sentence Ordering (BSO) is a binary classification task to determine the order of two sentences. The fact that BSO only has a pair of sentences as input makes it different from Sentence Position, where there is more context, and we hope that BSO can evaluate the ability of capturing local discourse coherence in the given sentence representations. The data comes from the same three domains as Sentence Position, and each instance is a pair of consecutive sentences.

Figure 6.5 shows an example from the arXiv domain of the Binary Sentence Ordering task. The order of the sentences in this instance is incorrect, as the "functions" are referenced before they are introduced. To detect the incorrect ordering in this example, the encoded representations need to be able to provide information about new and old information in each sentence.

To form the input when training classifiers, we concatenate the embeddings of both sentences with their element-wise difference: $[x_1, x_2, x_1 - x_2]$.

6.2.4 Discourse Coherence (DC)

Inspired by prior work on chat disentanglement [52, 53] and sentence clustering [219], we propose a sentence disentanglement task. The task is to determine whether a sequence of six sentences forms a coherent paragraph. We start with a coherent sequence of six sentences, then randomly replace one of the sentences (chosen uniformly among positions 2-5) with a sentence from another discourse. This task, which we call Discourse Coherence (DC), is a binary classification task and the datasets are balanced between positive and negative instances.

We use data from two domains for this task: Wikipedia and the Ubuntu IRC channel.[1] For Wikipedia, we begin by choosing a sequence of six sentences from a Wikipedia article. For purposes of choosing difficult distractor sentences, we use the Wikipedia categories of each document as an indication of its topic. To create a negative instance, we randomly sample a sentence from another document with a similar set of categories (measured by the percentage of overlapping categories). This sampled sentence replaces one of the six consecutive sentences in the original sequence. When splitting the train, development, and test sets, we ensure there are no overlapping documents among them.

Our proposed dataset differs from the sentence clustering task of [219] in that it preserves sentence order and does not anonymize or lemmatize words, because they play an important role in conveying information about discourse coherence.

For the Ubuntu domain, we use the human annotations of conversation thread structure from [101] to provide us with a coherent sequence of utterances. We filter out sentences by heuristic rules to avoid overly technical and unsolvable cases. The negative sentence is randomly picked from other conversations. Similarly, when splitting the train, development, and test sets, we ensure there are no overlapping conversations among them.

Figure 6.6 is an instance of the Wikipedia domain of the Discourse Coherence task. This

1. `https://irclogs.ubuntu.com/irclogs.ubuntu.com/`

1. It is possible he was the youngest of the family as the name "Sextus" translates to sixth in English implying he was the sixth of two living and three stillborn brothers.
2. According to Roman tradition, his rape of Lucretia was the precipitating event in the overthrow of the monarchy and the establishment of the Roman Republic.
3. Tarquinius Superbus was besieging Ardea, a city of the Rutulians.
4. The place could not be taken by force, and the Roman army lay encamped beneath the walls.
5. **He was soon elected to the Academy's membership (although he had to wait until 1903 to be elected to the Society of American Artists), and in 1883 he opened a New York studio, dividing his time for several years between Manhattan and Boston.**
6. As nothing was happening in the field, they mounted their horses to pay a surprise visit to their homes.

Figure 6.6: An example from the Wikipedia domain of the Discourse Coherence task.

instance is not coherent and the boldfaced text is from a different document. The incoherence can be found either by comparing characteristics of the entity being discussed or by the topic of the sentence group. Solving this task is non-trivial as it may require the ability to perform inference across multiple sentences.

In this task, we encode all sentences to vector representations and concatenate all of them ($[x_1, x_2, x_3, x_4, x_5, x_6]$) as input to the classification model. Note that in this task, we use a hidden layer of 2000 dimensions with sigmoid activation in the classification model, as this is necessary for the classifier to use features based on multiple inputs simultaneously given the simple concatenation as input. We could have developed richer ways to encode the input so that a linear classifier would be feasible (e.g., use the element-wise products of all pairs of sentence embeddings), but we wish to keep the input dimensionality of the classifier small enough that the classifier will be learnable given fixed sentence embeddings and limited training data.

6.2.5 Sentence Section Prediction (SSP)

The Sentence Section Prediction (SSP) task is defined as determining the section of a given sentence. The motivation behind this task is that sentences within certain sections typically

1. The theory behind the SVM and the naive Bayes classifier is explored.
2. This relocation of the active target may be repeated an arbitrary number of times.

Figure 6.7: Examples from Sentence Section Prediction.

Task	PDTB-E	PDTB-I	Ubuntu	RST-DT	Others
Train	9383	8693	5816	17051	10000
Dev.	3613	2972	1834	2045	4000
Test	3758	3024	2418	2308	4000

Table 6.1: Size of datasets in DiscoEval.

exhibit similar patterns because of the way people write coherent text. The pattern can be found based on connectives or specificity of a sentence. For example, "Empirically" is usually used in the abstract or introduction sections in scientific writing.

We construct the dataset from PeerRead [95], which consists of scientific papers from a variety of fields. The goal is to predict whether or not a sentence belongs to the Abstract section. After eliminating sentences that are too easy for the task (e.g., equations), we randomly sample sentences from the Abstract or from a section in the middle of a paper.[2] Figure 6.7 shows two sentences from this task, where the first sentence is more general and from an Abstract whereas the second is more specific and is from another section. In this task, the input to the classifier is simply the sentence embedding.

Table 6.1 shows the number of instances in each DiscoEval task introduced above.

6.3 Models and Learning Criteria

Having described DiscoEval, we now discuss methods for incorporating discourse information into sentence embedding training. All models in our experiments are composed of a single encoder and multiple decoders. The encoder, parameterized by a bidirectional Gated Recurrent Unit (BiGRU; 38), encodes the sentence, either in training or in evaluation of the

2. We avoid sentences from the Introduction or Conclusion sections to make the task more solvable.

downstream tasks, to a fixed-length vector representation (i.e., the average of the hidden states across positions).

The decoders take the aforementioned encoded sentence representation, and predict the targets we define in the sections below. We first introduce Neighboring Sentence Prediction, the loss for our baseline model. We then propose additional training losses to encourage our sentence embeddings to capture other context information.

6.3.1 Neighboring Sentence Prediction (NSP)

Similar to prior work on sentence embeddings [98, 72], we use an encoded sentence representation to predict its surrounding sentences. In particular, we predict the immediately preceding and succeeding sentences. All of our sentence embedding models use this loss. Formally, the loss is defined as

$$\text{NSP} = -\log p_\theta(s_{t-1}|s_t) - \log p_\phi(s_{t+1}|s_t)$$

where we parameterize p_θ and p_ϕ as separate feedforward neural networks and compute the log-probability of a target sentence using its bag-of-words representation.

6.3.2 Nesting Level (NL)

A table of contents serves as a high level description of an article, outlining its organizational structure. Wikipedia articles, for example, contain rich tables of contents with many levels of hierarchical structure. The "nesting level" of a sentence (i.e., how many levels deep it resides) provides information about its role in the overall discourse. To encode this information into our sentence representations, we introduce a discriminative loss to predict a sentence's nesting level in the table of contents:

$$\text{NL} = -\log p_\theta(l_t|s_t)$$

where l_t represents the nesting level of the sentence s_t and p_θ is parameterized by a feedforward neural network. Note that sentences within the same paragraph share the same nesting level. In Wikipedia, there are up to 7 nesting levels.

6.3.3 Sentence and Paragraph Position (SPP)

Similar to nesting level, we add a loss based on using the sentence representation to predict its position in the paragraph and in the article. The position of the sentence can be a strong indication of the relations between the topics of the current sentence and the topics in the entire article. For example, the first several sentences often cover the general topics to be discussed more thoroughly in the following sentences. To encourage our sentence embeddings to capture such information, we define a position prediction loss

$$\text{SPP} = -\log p_\theta(sp_t|s_t) - \log p_\phi(pp_t|s_t)$$

where sp_t is the sentence position of s_t within the current paragraph and pp_t is the position of the current paragraph in the whole document.

6.3.4 Section and Document Title (SDT)

Unlike the previous position-based losses, this loss makes use of section and document titles, which gives the model more direct access to the topical information at different positions in the document. The loss is defined as

$$\text{SDT} = -\log p_\theta(st_t|s_t) - \log p_\phi(dt_t|s_t)$$

Where st_t is the section title of sentence s_t, dt_t is the document title of sentence s_t, and p_θ and p_ϕ are two different bag-of-words decoders.

	SentEval			
	USS	SSS	SC	Probing
Skip-thought	41.7	81.2	78.4	70.1
InferSent	**63.4**	**83.3**	79.7	71.8
DisSent	50.0	79.2	80.5	74.0
ELMo	60.9	77.6	80.8	74.7
BERT-Base	30.1	66.3	81.4	73.9
BERT-Large	43.6	70.7	**83.4**	**75.0**
Baseline (NSP)	57.8	77.1	77.0	70.6
+ SDT	59.0	77.3	76.8	69.7
+ SPP	56.0	77.5	77.4	70.7
+ NL	56.7	78.2	77.2	70.6
+ SPP + NL	55.4	76.7	77.0	70.4
+ SDT + NL	58.5	76.9	77.2	70.2
+ SDT +SPP	58.4	77.4	76.6	70.2
ALL	58.8	76.3	77.0	70.2

Table 6.2: Results for SentEval. The highest number in each column is boldfaced.

6.4 Experiments

6.4.1 Setup

We train our models on Wikipedia as it is a knowledge rich textual resource and has consistent structures over all documents. Details on hyperparameters are in the supplementary material. When evaluating on DiscoEval, we encode sentences with pretrained sentence encoders. Following SentEval, we freeze the sentence encoders and only learn the parameters of the downstream classifier. The "Baseline" row in Table 6.2 and Table 6.3 are embeddings trained with only the NSP loss. The subsequent rows are trained with extra losses defined in Section 6.3 in addition to the NSP loss.

Additionally, we benchmark several popular pretrained sentence encoders on DiscoEval,

	DiscoEval							
	SP	BSO	DC	SSP	PDTB-E	PDTB-I	RST-DT	avg.
Skip-thought	47.5	64.6	55.2	77.5	39.3	40.2	**59.7**	54.8
InferSent	45.8	62.9	56.3	62.2	37.3	38.8	52.3	50.8
DisSent	47.7	64.9	54.8	62.2	42.2	40.7	57.8	52.9
ELMo	47.8	65.6	**60.7**	79.0	41.3	41.8	57.5	56.2
BERT-Base	53.1	68.5	58.9	80.3	41.9	42.4	58.8	57.7
BERT-Large	**53.8**	**69.3**	59.6	**80.4**	**44.3**	**43.6**	59.1	**58.6**
Baseline (NSP)	47.3	63.8	61.0	77.8	36.5	39.1	56.7	54.6
+ SDT	45.8	62.9	60.3	78.0	36.6	39.1	55.7	54.1
+ SPP	48.4	65.3	60.2	78.4	38.1	39.9	56.4	55.2
+ NL	46.9	64.0	61.0	78.9	37.6	39.9	56.5	55.0
+ SPP + NL	48.5	64.7	59.9	78.9	37.8	40.5	56.7	55.3
+ SDT + NL	46.1	63.0	60.8	78.1	36.7	38.1	56.2	54.1
+ SDT +SPP	46.5	63.9	60.4	77.6	35.2	38.6	56.3	54.1
ALL	46.1	63.7	60.0	78.6	36.3	37.6	55.3	53.9

Table 6.3: Results for DiscoEval. The highest number in each column is boldfaced.

including Skip-thought,[3] InferSent [41],[4] DisSent [154],[5] ELMo,[6] and BERT.[7] For ELMo, we use the averaged vector of all three layers and time steps as the sentence representations. For BERT, we use the averaged vector at the position of the "[CLS]" token across all layers. We also evaluate per-layer performance for both models in Section 6.5.

When reporting results for SentEval, we compute the averaged Pearson correlations for Semantic Textual Similarity tasks from 2012 to 2016 [6, 7, 4, 3, 5]. We refer to the average as unsupervised semantic similarity (USS) since those tasks do not require training data. We compute the averaged results for the STS Benchmark [24], textual entailment, and semantic relatedness [136] and refer to the average as supervised semantic similarity (SSS). We compute the average accuracy for movie review [162]; customer review [80]; opinion

3. https://github.com/ryankiros/skip-thoughtsgithub.com/ryankiros/skip-thoughts

4. https://github.com/facebookresearch/InferSentgithub.com/facebookresearch/InferSent

5. https://github.com/windweller/DisExtractgithub.com/windweller/DisExtract

6. https://github.com/allenai/allennlpgithub.com/allenai/allennlp

7. https://github.com/huggingface/pytorch-pretrained-BERTgithub.com/huggingface/pytorch-pretrained-BERT

polarity [224]; subjectivity classification [161]; Stanford sentiment treebank [195]; question classification [114]; and paraphrase detection [49], and refer to it as sentence classification (SC). For the rest of the linguistic probing tasks [42], we report the average accuracy and report it as "Probing".

6.4.2 Results

Table 6.2 and table 6.3 shows the experiment results over all SentEval and DiscoEval tasks. Different models and training signals have complex effects when performing various downstream tasks. We summarize our findings below:

- On DiscoEval, Skip-thought performs best on RST-DT. DisSent performs strongly for PDTB tasks but it requires discourse markers from PDTB for generating training data. BERT has the highest average by a large margin, but ELMo has competitive performance on multiple tasks.
- The NL or SPP loss alone has complex effects across tasks in DiscoEval, but when they are combined, the model achieves the best performance, outperforming our baseline by 0.7% on average. In particular, it yields 40.5% accuracy on PDTB-I, outperforming Skip-thought by 0.3%. This is presumably caused by the differing, yet complementary, effects of these two losses (NL and SPP).
- The SDT loss generally hurts performance on DiscoEval, especially on the position-related tasks (SP, BSO). This can be explained by the notion that consecutive sentences in the same section are encouraged to have the same sentence representations when using the SDT loss. However, the SP and BSO tasks involve differentiating neighboring sentences in terms of their position and ordering information.
- On SentEval, SDT is most helpful for the USS tasks, presumably because it provides the most direct information about the topic of each sentence, which is a component of semantic similarity. SDT helps slightly on the SSS tasks. NL gives the biggest improvement in SSS.

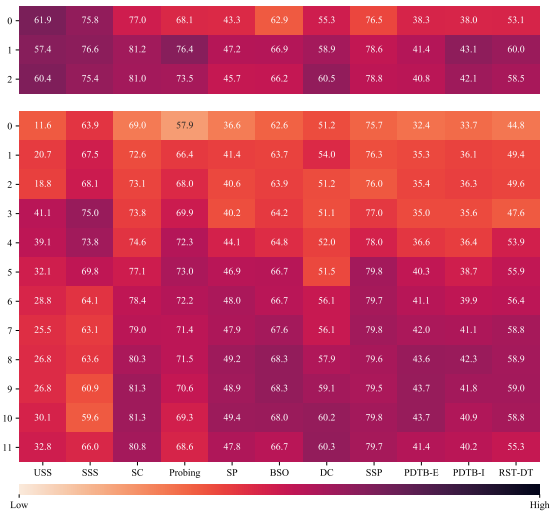


Figure 6.8: Heatmap for individual hidden layers of BERT-Base (lower part) and ELMo (upper part).

	ELMo	BERT-Base
SentEval	0.8	5.0
DiscoEval	1.3	8.9

Table 6.4: Average of the layer number for the best layers in SentEval and DiscoEval.

- In comparing BERT to ELMo and Skip-thought to InferSent on DiscoEval, we can see the benefit of adding information about neighboring sentences. Our proposed training objectives show complementary improvements over NSP, which suggests that they can potentially benefit these pretrained representations.

6.5 Analysis

Per-Layer analysis. To investigate the performance of individual hidden layers, we evaluate ELMo and BERT on both SentEval and DiscoEval using each hidden layer. For ELMo, we use the averaged vector from the targeted layer. For BERT-Base, we use the vector from the position of the "[CLS]" token. Figure 6.8 shows the heatmap of performance for individual hidden layers. We note that for better visualization, colors in each column are

Baseline w/o hidden layer	52.0
Baseline w/ hidden layer	61.0

Table 6.5: Accuracies with baseline encoder on Discourse Coherence task, with or without a hidden layer in the classifier.

	Sentence Position			Binary Sentence Ordering			Discourse Coherence	
Human	77.3			84.7			87.0	
BERT-Large	53.8			69.3			59.6	
	Wiki	arXiv	ROC	Wiki	arXiv	ROC	Wiki	Ubuntu
Human	84.0	76.0	94.0	64.0	72.0	96.0	98.0	74.0
BERT-Large	50.7	47.3	63.4	70.4	66.8	70.8	65.1	54.2

Table 6.6: Accuracies (%) for a human annotator and BERT-Large on Sentence Position, Binary Sentence Ordering, and Discourse Coherence tasks.

standardized. On SentEval, BERT-Base performs better with shallow layers on USS, SSS, and Probing (though not on SC), but on DiscoEval, the results using BERT-Base gradually increase with deeper layers. To evaluate this phenomenon quantitatively, we compute the average of the layer number for the best layers for both ELMo and BERT-Base and show it in Table 6.4. From the table, we can see that DiscoEval requires deeper layers to achieve better performance. We assume this is because deeper layers can capture higher-level structure, which aligns with the information needed to solve the discourse tasks.

DiscoEval architectures. In all DiscoEval tasks except DC, we use no hidden layer in the neural architectures, following the example of SentEval. However, some tasks are unsolvable with this simple architecture. In particular, the DC tasks have low accuracies with all models unless a hidden layer is used. As shown in Table 6.5, when adding a hidden layer of 2000 to this task, the performance on DC improves dramatically. This shows that DC requires more complex comparison and inference among input sentences. Our human evaluation below on DC also shows that human accuracies exceed those of the classifier based on sentence embeddings by a large margin.

Random	20
Baseline w/o context	43.2
Baseline w/ context	47.3

Table 6.7: Accuracies (%) for baseline encoder on Sentence Position task when using downstream classifier with or without context.

Human Evaluation. We conduct a human evaluation on the Sentence Position, Binary Sentence Ordering, and Discourse Coherence datasets. A native English speaker was provided with 50 examples per domain for these tasks. While the results in Table 6.6 show that the overall human accuracies exceed those of the classifier based on BERT-Large by a large margin, we observe that within some specific domains, for example Wiki in BSO, BERT-Large demonstrates very strong performance.

Does context matter in Sentence Position? In the SP task, the inputs are the target sentence together with 4 surrounding sentences. We study the effect of removing the surrounding 4 sentences, i.e., only using the target sentence to predict its position from the start of the paragraph.

Table 6.7 shows the comparison of the baseline model performance on Sentence Position with or without the surrounding sentences and a random baseline. Since our baseline model is already trained with NSP, it is expected to see improvements over a random baseline. The further improvement from using surrounding sentences demonstrates that the context information is helpful in determining the sentence position.

6.6 Summary

In this chapter, we proposed DiscoEval, a test suite of tasks to evaluate discourse-related knowledge encoded in pretrained sentence representations. We also proposed a variety of training objectives to strengthen encoders' ability to incorporate discourse information. We benchmarked several pretrained sentence encoders and demonstrated the effects of the pro-

posed training objectives on different tasks. While our learning criteria showed benefit on certain classes of tasks, our hope is that the DiscoEval evaluation suite can inspire additional research in capturing broad discourse context in fixed-dimensional sentence embeddings.

CHAPTER 7

CONCLUSION

We summarize our contributions in this chapter to conclude this thesis, and discuss future works.

7.1 Summary of Thesis

This thesis made the following contributions to learning text representations and evaluations:

- We build weakly supervised text classifiers with text and annotated categories as training resources. These classifiers are evaluated on CatEval, a collection of topical and sentiment classification tasks.

- Using Wikipedia category structure data, we pretrain text representations for natural language inference tasks.

- We add entity knowledge into text representations by leveraging Wikipedia documents and hyperlinks information. We also propose a standard benchmark suite EntEval to evaluate entity embedding.

- We train discourse knowledge injected sentence representations using Wikipedia text and document discourse structures. We also build a standard test suite DiscoEval to test the effectiveness of different sentence embedding.

7.2 Future Work

In extension of this thesis, the following directions can be explored.

Pretraining Text Representations with Knowledge More knowledge can be explored to be embedded into text representations. For instance, a knowledge graph with both structural links and text descriptions can be a good resource for learning text representations with rich knowledge. Some recent work [234] explored this direction.

Some work shows that pretrained models contain commonsense knowledge [207].

Universal topic classifiers In addition to the NatCat dataset, more datasets with weakly supervised signals can be explored. For instance, titles of articles could be considered as signals for topic classifications. The same applies for web pages with page titles, headers, or tweets with hashtags, or any other documents with naturally annotated tags. Harvesting such free resources online for document classification is a direction worth trying.

Combing multiple training signals In this thesis we considered multiple knowledge resources to be added into text representations. It would be natural to combine all the knowledge into a single model.

A single BERT based model could be a natural way to unify all training signals. However, different training losses may be conflicting each other, so more research is needed regarding how to properly combine all the knowledge.

In fact, for EntEval, we tried to fine-tune a BERT encoder with bag of words decoder, but the performance drops after fine-tuning. It is trivial to build a BERT based entity encoder from our experience.

REFERENCES

[1] Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *ICLR*, 2017.

[2] Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *Proceedings of ICLR*, 2017.

[3] Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Inigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 252–263. Association for Computational Linguistics, 2015.

[4] Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. Semeval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 81–91. Association for Computational Linguistics, 2014.

[5] Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511. Association for Computational Linguistics, 2016.

[6] Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. Semeval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393. Association for Computational Linguistics, 2012.

[7] Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. *sem 2013 shared task: Semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43. Association for Computational Linguistics, 2013.

[8] Gabor Angeli and Christopher D. Manning. NaturalLI: Natural logic inference for common sense reasoning. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 534–545, Doha, Qatar, October 2014. Association for Computational Linguistics.

[9] Collin Baker. FrameNet: A knowledge base for natural language processing. In *Proceedings of Frame Semantics in NLP: A Workshop in Honor of Chuck Fillmore (1929-2014)*, pages 1–5, Baltimore, MD, USA, June 2014. Association for Computational Linguistics.

[10] Mohit Bansal, David Burkett, Gerard de Melo, and Dan Klein. Structured learning for taxonomy induction with belief propagation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1041–1051, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[11] Roy Bar Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. The second PASCAL recognising textual entailment challenge. 2006.

[12] Marco Baroni, Raffaella Bernardi, Ngoc-Quynh Do, and Chung-chieh Shan. Entailment above the word level in distributional semantics. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 23–32, Avignon, France, April 2012. Association for Computational Linguistics.

[13] Regina Barzilay and Mirella Lapata. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34, 2008.

[14] Yonatan Belinkov, Lluís Màrquez, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. Evaluating layers of representation in neural machine translation on part-of-speech and semantic tagging tasks. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1–10, Taipei, Taiwan, November 2017. Asian Federation of Natural Language Processing.

[15] Luisa Bentivogli, Ido Dagan, Hoa Trang Dang, Danilo Giampiccolo, and Bernardo Magnini. The fifth PASCAL recognizing textual entailment challenge. 2009.

[16] Jonathan Berant, Noga Alon, Ido Dagan, and Jacob Goldberger. Efficient global learning of entailment graphs. *Computational Linguistics*, 41(2):221–263, June 2015.

[17] Rahul Bhagat, Patrick Pantel, and Eduard Hovy. LEDIR: An unsupervised algorithm for learning directionality of inference rules. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 161–170, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[18] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. AcM, 2008.

[19] Antoine Bosselut, Asli Celikyilmaz, Xiaodong He, Jianfeng Gao, Po-Sen Huang, and Yejin Choi. Discourse-aware neural rewards for coherent text generation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 173–184, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[20] Zied Bouraoui, Jose Camacho-Collados, and Steven Schockaert. Inducing relational knowledge from bert. *arXiv preprint arXiv:1911.12753*, 2019.

[21] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[22] Samuel Broscheit. Investigating entity knowledge in BERT with simple neural end-to-end entity linking. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 677–685, Hong Kong, China, November 2019. Association for Computational Linguistics.

[23] Lynn Carlson, Daniel Marcu, and Mary Ellen Okurovsky. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue*, 2001.

[24] Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14. Association for Computational Linguistics, 2017.

[25] Ming-Wei Chang, Lev Ratinov, Dan Roth, and Vivek Srikumar. Importance of semantic representation: Dataless classification. In *AAAI*, 2008.

[26] Hong Chen, Zhenhua Fan, Hao Lu, Alan Yuille, and Shu Rong. PreCo: A large-scale dataset in preschool vocabulary for coreference resolution. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 172–181, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[27] Mingda Chen, Zewei Chu, Yang Chen, Karl Stratos, and Kevin Gimpel. EntEval: A holistic evaluation benchmark for entity representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 421–433, Hong Kong, China, November 2019. Association for Computational Linguistics.

[28] Mingda Chen, Zewei Chu, and Kevin Gimpel. Evaluation benchmarks and learning criteria for discourse-aware sentence representations. In *Proc. of EMNLP*, 2019.

[29] Mingda Chen, Zewei Chu, and Kevin Gimpel. Evaluation benchmarks and learning criteria for discourse-aware sentence representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 649–662, Hong Kong, China, November 2019. Association for Computational Linguistics.

[30] Mingda Chen, Zewei Chu, Karl Stratos, and Kevin Gimpel. Enteval: A holistic evaluation benchmark for entity representations. In *Proc. of EMNLP*, 2019.

[31] Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Diana Inkpen, and Si Wei. Neural natural language inference models enhanced with external knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2406–2417, 2018.

[32] Sihao Chen, Daniel Khashabi, Wenpeng Yin, Chris Callison-Burch, and Dan Roth. Seeing things from a different angle:discovering diverse perspectives about claims. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 542–557, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[33] Xinchi Chen, Xipeng Qiu, and Xuanjing Huang. Neural sentence ordering. *arXiv preprint arXiv:1607.06952*, 2016.

[34] Xingyuan Chen, Yunqing Xia, Peng Jin, and John Carroll. Dataless text classification with descriptive LDA. In *AAAI*, 2015.

[35] 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. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, 2014.

[36] Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. Ultra-fine entity typing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 87–96, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[37] Zewei Chu, Jing Chen, Kevin Gimpel, Manaal Faruqui, Miaosen Wang, Mingda Chen, and Xiance Si. How to ask better questions? a large-scale multi-domain dataset for rewriting ill-formed questions. In *AAAI*, 2020.

[38] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[39] Elizabeth Clark, Yangfeng Ji, and Noah A. Smith. Neural text generation in stories using entity representations as context. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2250–2260, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[40] Alexis Conneau and Douwe Kiela. SentEval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, Miyazaki, Japan, May 2018. European Languages Resources Association (ELRA).

[41] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680. Association for Computational Linguistics, 2017.

[42] Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[43] Baiyun Cui, Yingming Li, Ming Chen, and Zhongfei Zhang. Deep attentive sentence ordering network. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4340–4349. Association for Computational Linguistics, 2018.

[44] Ido Dagan, Oren Glickman, and Bernardo Magnini. The PASCAL recognising textual entailment challenge. In *Machine learning challenges. evaluating predictive uncertainty, visual object classification, and recognising tectual entailment*, pages 177–190. Springer, 2006.

[45] Yann N Dauphin, Gokhan Tur, Dilek Hakkani-Tur, and Larry Heck. Zero-shot learning for semantic utterance classification. *arXiv preprint arXiv:1401.0509*, 2013.

[46] Luciano Del Corro, Abdalghani Abujabal, Rainer Gemulla, and Gerhard Weikum. FINET: Context-aware fine-grained named entity typing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 868–878, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[47] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

[48] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings*

of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[49] Bill Dolan, Chris Quirk, and Chris Brockett. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*, pages 350–356, Geneva, Switzerland, Aug 23–Aug 27 2004.

[50] Greg Durrett and Dan Klein. Easy victories and uphill battles in coreference resolution. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1971–1982, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

[51] Greg Durrett and Dan Klein. A joint model for entity analysis: Coreference, typing, and linking. *Transactions of the Association for Computational Linguistics*, 2:477–490, 2014.

[52] Micha Elsner and Eugene Charniak. You talking to me? a corpus and algorithm for conversation disentanglement. In *Proceedings of ACL-08: HLT*, pages 834–842, Columbus, Ohio, June 2008. Association for Computational Linguistics.

[53] Micha Elsner and Eugene Charniak. Disentangling chat. *Computational Linguistics*, 36(3):389–409, 2010.

[54] Allyson Ettinger. What bert is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *arXiv preprint arXiv:1907.13528*, 2019.

[55] Geli Fei and Bing Liu. Breaking the closed world assumption in text classification. In *NAACL*, 2016.

[56] Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. Language, Speech, and Communication. MIT Press, Cambridge, MA, 1998.

[57] Elisa Ferracane, Greg Durrett, Junyi Jessy Li, and Katrin Erk. Evaluating discourse in structured text representations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 646–653, Florence, Italy, July 2019. Association for Computational Linguistics.

[58] Matthew Francis-Landau, Greg Durrett, and Dan Klein. Capturing semantic similarity for entity linking with convolutional neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1256–1261, San Diego, California, June 2016. Association for Computational Linguistics.

[59] Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI*, 2007.

[60] Zhe Gan, Yunchen Pu, Ricardo Henao, Chunyuan Li, Xiaodong He, and Lawrence Carin. Learning generic sentence representations using convolutional neural networks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2390–2400, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[61] Octavian-Eugen Ganea and Thomas Hofmann. Deep joint entity disambiguation with local neural attention. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2619–2629, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[62] Shima Gerani, Yashar Mehdad, Giuseppe Carenini, Raymond T. Ng, and Bita Nejat. Abstractive summarization of product reviews using discourse structure. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1602–1613, Doha, Qatar, October 2014. Association for Computational Linguistics.

[63] Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*, pages 1–9. Association for Computational Linguistics, 2007.

[64] Max Glockner, Vered Shwartz, and Yoav Goldberg. Breaking NLI systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[65] Barbara J. Grosz, Scott Weinstein, and Aravind K. Joshi. Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics*, 21(2), 1995.

[66] Zhaochen Guo and Denilson Barbosa. Robust entity linking via random walks. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '14, pages 499–508, New York, NY, USA, 2014. ACM.

[67] Nitish Gupta, Sameer Singh, and Dan Roth. Entity linking via joint encoding of types, descriptions, and context. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2681–2690, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[68] Francisco Guzmán, Shafiq Joty, Lluís Màrquez, and Preslav Nakov. Using discourse structure improves machine translation evaluation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 687–698, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[69] He He, Anusha Balakrishnan, Mihail Eric, and Percy Liang. Learning symmetric collaborative dialogue agents with dynamic knowledge graph embeddings. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1766–1776, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[70] Zhengyan He, Shujie Liu, Mu Li, Ming Zhou, Longkai Zhang, and Houfeng Wang. Learning entity representation for entity disambiguation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 30–34, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[71] Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *COLING 1992 Volume 2: The 15th International Conference on Computational Linguistics*, 1992.

[72] Felix Hill, Kyunghyun Cho, and Anna Korhonen. Learning distributed representations of sentences from unlabelled data. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1367–1377. Association for Computational Linguistics, 2016.

[73] Felix Hill, KyungHyun Cho, and Anna Korhonen. Learning sentence representations from unlabelled data. In *NAACL-HLT*, 2016.

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

[75] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. In *Neural Computation*, 1997.

[76] Johannes Hoffart, Stephan Seufert, Dat Ba Nguyen, Martin Theobald, and Gerhard Weikum. Kore: keyphrase overlap relatedness for entity disambiguation. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 545–554. ACM, 2012.

[77] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust disambiguation of named entities in text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 782–792. Association for Computational Linguistics, 2011.

[78] Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017.

[79] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In *ACL*, 2018.

[80] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM, 2004.

[81] Hongzhao Huang, Larry Heck, and Heng Ji. Leveraging deep neural networks and knowledge graphs for entity disambiguation. *arXiv preprint arXiv:1504.07678*, 2015.

[82] Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. On using very large target vocabulary for neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1–10, Beijing, China, July 2015. Association for Computational Linguistics.

[83] Yacine Jernite, Samuel R Bowman, and David Sontag. Discourse-based objectives for fast unsupervised sentence representation learning. *arXiv preprint arXiv:1705.00557*, 2017.

[84] Yangfeng Ji and Jacob Eisenstein. Representation learning for text-level discourse parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13–24, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[85] Yangfeng Ji and Jacob Eisenstein. One vector is not enough: Entity-augmented distributed semantics for discourse relations. *Transactions of the Association for Computational Linguistics*, 3:329–344, 2015.

[86] Yangfeng Ji and Noah A. Smith. Neural discourse structure for text categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 996–1005, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[87] Yangfeng Ji, Chenhao Tan, Sebastian Martschat, Yejin Choi, and Noah A. Smith. Dynamic entity representations in neural language models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1830–1839, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[88] Jyun-Yu Jiang, Francine Chen, Yan-Ying Chen, and Wei Wang. Learning to disentangle interleaved conversational threads with a siamese hierarchical network and similarity ranking. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1812–1822. Association for Computational Linguistics, 2018.

[89] Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In *ECML*, 1998.

[90] Rie Johnson and Tong Zhang. Semi-supervised convolutional neural networks for text categorization via region embedding. In *NIPS*, 2015.

[91] Rie Johnson and Tong Zhang. Convolutional neural networks for text categorization: Shallow word-level vs. deep character-level. In *arXiv*, 2016.

[92] Rie Johnson and Tong Zhang. Deep pyramid convolutional neural networks for text categorization. In *ACL*, 2017.

[93] Daniel Jurafsky and James H. Martin. *Speech and Language Processing (2Nd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2009.

[94] Nal Kalchbrenner and Phil Blunsom. Recurrent convolutional neural networks for discourse compositionality. In *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*, pages 119–126. Association for Computational Linguistics, 2013.

[95] Dongyeop Kang, Waleed Ammar, Bhavana Dalvi, Madeleine van Zuylen, Sebastian Kohlmeier, Eduard Hovy, and Roy Schwartz. A dataset of peer reviews (peerread): Collection, insights and nlp applications. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1647–1661. Association for Computational Linguistics, 2018.

[96] Ben Kantor and Amir Globerson. Coreference resolution with entity equalization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 673–677, Florence, Italy, July 2019. Association for Computational Linguistics.

[97] Yoon Kim. Convolutional neural networks for sentence classification. In *EMNLP*, 2014.

[98] Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Skip-thought vectors. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, pages 3294–3302, Cambridge, MA, USA, 2015. MIT Press.

[99] Roman Klinger et al. An analysis of annotated corpora for emotion classification in text. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2104–2119, 2018.

[100] Lili Kotlerman, Ido Dagan, Idan Szpektor, and Maayan Zhitomirsky-Geffet. Directional distributional similarity for lexical expansion. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 69–72, Suntec, Singapore, August 2009. Association for Computational Linguistics.

[101] Jonathan K. Kummerfeld, Sai R. Gouravajhala, Joseph J. Peper, Vignesh Athreya, Chulaka Gunasekara, Jatin Ganhotra, Siva Sankalp Patel, Lazaros C Polymenakos, and

Walter Lasecki. A large-scale corpus for conversation disentanglement. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3846–3856, Florence, Italy, July 2019. Association for Computational Linguistics.

[102] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

[103] Ken Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 331–339, 1995.

[104] Phong Le and Ivan Titov. Improving entity linking by modeling latent relations between mentions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1595–1604, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[105] Phong Le and Ivan Titov. Boosting entity linking performance by leveraging unlabeled documents. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1935–1945, Florence, Italy, July 2019. Association for Computational Linguistics.

[106] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196, 2014.

[107] Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[108] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195, 2015.

[109] Omer Levy, Ido Dagan, and Jacob Goldberger. Focused entailment graphs for open IE propositions. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 87–97, Ann Arbor, Michigan, June 2014. Association for Computational Linguistics.

[110] Omer Levy, Steffen Remus, Chris Biemann, and Ido Dagan. Do supervised distributional methods really learn lexical inference relations? In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 970–976, Denver, Colorado, May–June 2015. Association for Computational Linguistics.

[111] Chenliang Li, Jian Xing, Aixin Sun, and Zongyang Ma. Effective document labeling with very few seed words: A topic model approach. In *CIKM*, 2016.

[112] Jiwei Li and Dan Jurafsky. Neural net models of open-domain discourse coherence. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 198–209. Association for Computational Linguistics, 2017.

[113] Xiang Li, Aynaz Taheri, Lifu Tu, and Kevin Gimpel. Commonsense knowledge base completion. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1445–1455, Berlin, Germany, August 2016. Association for Computational Linguistics.

[114] Xin Li and Dan Roth. Learning question classifiers. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics, 2002.

[115] Yuezhang Li, Ronghuo Zheng, Tian Tian, Zhiting Hu, Rahul Iyer, and Katia Sycara. Joint embedding of hierarchical categories and entities for concept categorization and dataless classification. *COLING*, 2016.

[116] Dekang Lin and Patrick Pantel. Discovery of inference rules for question-answering. *Nat. Lang. Eng.*, 7(4):343–360, December 2001.

[117] Xiang Lin, Shafiq Joty, Prathyusha Jwalapuram, and M Saiful Bari. A unified linear-time framework for sentence-level discourse parsing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4190–4200, Florence, Italy, July 2019. Association for Computational Linguistics.

[118] Ziheng Lin, Min-Yen Kan, and Hwee Tou Ng. Recognizing implicit discourse relations in the penn discourse treebank. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 343–351. Association for Computational Linguistics, 2009.

[119] Xiao Ling, Sameer Singh, and Daniel S. Weld. Design challenges for entity linking. *Transactions of the Association for Computational Linguistics*, 3:315–328, 2015.

[120] Jiangming Liu, Shay B. Cohen, and Mirella Lapata. Discourse representation structure parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 429–439, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[121] Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019.

[122] Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and*

Short Papers), pages 1073–1094, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[123] Yang Liu and Mirella Lapata. Learning structured text representations. *Transactions of the Association for Computational Linguistics*, 6:63–75, 2018.

[124] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[125] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

[126] Robert Logan, Nelson F. Liu, Matthew E. Peters, Matt Gardner, and Sameer Singh. Barack's wife hillary: Using knowledge graphs for fact-aware language modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5962–5971, Florence, Italy, July 2019. Association for Computational Linguistics.

[127] Lajanugen Logeswaran, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Jacob Devlin, and Honglak Lee. Zero-shot entity linking by reading entity descriptions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3449–3460, Florence, Italy, July 2019. Association for Computational Linguistics.

[128] Lajanugen Logeswaran and Honglak Lee. An efficient framework for learning sentence representations. In *International Conference on Learning Representations*, 2018.

[129] Lajanugen Logeswaran, Honglak Lee, and Dragomir Radev. Sentence ordering and coherence modeling using recurrent neural networks, 2016.

[130] Teng Long, Emmanuel Bengio, Ryan Lowe, Jackie Chi Kit Cheung, and Doina Precup. World knowledge for reading comprehension: Rare entity prediction with hierarchical lstms using external descriptions. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 825–834, 2017.

[131] Edward Loper and Steven Bird. NLTK: The Natural Language Toolkit, 2002. `http://nltk.sourceforge.net/`.

[132] Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. The Ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 285–294, Prague, Czech Republic, September 2015. Association for Computational Linguistics.

[133] Yukun Ma, Erik Cambria, and Sa Gao. Label embedding for zero-shot fine-grained named entity typing. *COLING*, 2016.

[134] William C Mann and Sandra A Thompson. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281, 1988.

[135] Daniel Marcu. *The Theory and Practice of Discourse Parsing and Summarization*. MIT Press, Cambridge, MA, USA, 2000.

[136] Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. SemEval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, 2014.

[137] Pedro Henrique Martins, Zita Marinho, and André F. T. Martins. Joint learning of named entity recognition and entity linking. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 190–196, Florence, Italy, July 2019. Association for Computational Linguistics.

[138] Stephen Mayhew, Tatiana Tsygankova, Francesca Marini, Zihan Wang, Jane Lee, Xiaodong Yu, Xingyu Fu, Weijia Shi, Zian Zhao, Wenpeng Yin, Karthikeyan K, Jamaal Hay, Michael Shur, Jennifer Sheffield, and Dan Roth. University of pennsylvania lorehlt 2019 submission. 2019.

[139] Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6294–6305, 2017.

[140] Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. Learned in translation: Contextualized word vectors. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6294–6305. Curran Associates, Inc., 2017.

[141] Paul McNamee, Rion Snow, Patrick Schone, and James Mayfield. Learning named entity hyponyms for question answering. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II*, 2008.

[142] Shikib Mehri and Giuseppe Carenini. Chat disentanglement: Identifying semantic reply relationships with random forests and recurrent neural networks. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 615–623, Taipei, Taiwan, November 2017. Asian Federation of Natural Language Processing.

[143] Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. Weakly-supervised hierarchical text classification. In *AAAI*, 2019.

[144] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages

2381–2391, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[145] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[146] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *ACL*, 2009.

[147] Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California, June 2016. Association for Computational Linguistics.

[148] James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. Explainable prediction of medical codes from clinical text. *arXiv preprint arXiv:1802.05695*, 2018.

[149] Shikhar Murty, Patrick Verga, Luke Vilnis, and Andrew McCallum. Finer grained entity typing with typenet. *arXiv preprint arXiv:1711.05795*, 2017.

[150] Jinseok Nam, Eneldo Loza Mencía, and Johannes Fürnkranz. All-in text: Learning document, label, and word representations jointly. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[151] Karthik Narasimhan and Regina Barzilay. Machine comprehension with discourse relations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1253–1262, Beijing, China, July 2015. Association for Computational Linguistics.

[152] Vivi Nastase, Michael Strube, Benjamin Boerschinger, Caecilia Zirn, and Anas Elghafari. WikiNet: A very large scale multi-lingual concept network. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May 2010. European Languages Resources Association (ELRA).

[153] Denis Newman-Griffis, Albert M Lai, and Eric Fosler-Lussier. Jointly embedding entities and text with distant supervision. In *Proceedings of The Third Workshop on Representation Learning for NLP*, pages 195–206, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[154] Allen Nie, Erin Bennett, and Noah Goodman. DisSent: Learning sentence representations from explicit discourse relations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4497–4510, Florence, Italy, July 2019. Association for Computational Linguistics.

[155] Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial nli: A new benchmark for natural language understanding, 2019.

[156] Kamal Nigam, Andrew Mccallum, Sebastian Thrun, and Tom Mitchell. Text classification from labeled and unlabeled documents using em. In *Machine Learning*, 2000.

[157] Rasha Obeidat, Xiaoli Fern, Hamed Shahbazi, and Prasad Tadepalli. Description-based zero-shot fine-grained entity typing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 807–814, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[158] Yasumasa Onoe and Greg Durrett. Learning to denoise distantly-labeled data for entity typing. In *NAACL-HLT*, 2019.

[159] Boyuan Pan, Yazheng Yang, Zhou Zhao, Yueting Zhuang, Deng Cai, and Xiaofei He. Discourse marker augmented network with reinforcement learning for natural language inference. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 989–999, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[160] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 2009.

[161] Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*, page 271. Association for Computational Linguistics, 2004.

[162] Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 115–124. Association for Computational Linguistics, 2005.

[163] Raghavendra Pappagari, Piotr Żelasko, Jesús Villalba, Yishay Carmiel, and Najim Dehak. Hierarchical transformers for long document classification. *arXiv preprint arXiv:1910.10781*, 2019.

[164] Marius Pasca and Benjamin Van Durme. What you seek is what you get: Extraction of class attributes from query logs. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07)*, pages 2832–2837, 2007.

[165] Ellie Pavlick, Johan Bos, Malvina Nissim, Charley Beller, Benjamin Van Durme, and Chris Callison-Burch. Adding semantics to data-driven paraphrasing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1512–1522, Beijing, China, July 2015. Association for Computational Linguistics.

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

[167] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[168] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014.

[169] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics, 2018.

[170] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, 2018.

[171] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[172] Matthew Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. Dissecting contextual word embeddings: Architecture and representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[173] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *NAACL*, 2018.

[174] Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. Knowledge enhanced contextual word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 43–54, Hong Kong, China, November 2019. Association for Computational Linguistics.

[175] Matthew E. Peters, Sebastian Ruder, and Noah A. Smith. To tune or not to tune? adapting pretrained representations to diverse tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 7–14, Florence, Italy, August 2019. Association for Computational Linguistics.

[176] Adam Poliak, Aparajita Haldar, Rachel Rudinger, J. Edward Hu, Ellie Pavlick, Aaron Steven White, and Benjamin Van Durme. Collecting diverse natural language inference problems for sentence representation evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 67–81, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[177] Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. The penn discourse treebank 2.0. In *In Proceedings of LREC*, 2008.

[178] Raul Puri and Bryan Catanzaro. Zero-shot text classification with generative language models. *arXiv preprint arXiv:1912.10165*, 2019.

[179] Maxim Rabinovich and Dan Klein. Fine-grained entity typing with high-multiplicity assignments. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 330–334, 2017.

[180] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/language-unsupervised/language_ understanding_paper. pdf*, 2018.

[181] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8), 2019.

[182] Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. Explain yourself! leveraging language models for commonsense reasoning. *arXiv preprint arXiv:1906.02361*, 2019.

[183] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[184] Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 74–84, Atlanta, Georgia, June 2013. Association for Computational Linguistics.

[185] Anthony Rios and Ramakanth Kavuluru. Few-shot and zero-shot multi-label learning for structured label spaces. In *EMNLP*, 2018.

[186] Evan Sandhaus. The New York Times Annotated Corpus, 2008.

[187] Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. Socialiqa: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*, 2019.

[188] Xing Shi, Inkit Padhi, and Kevin Knight. Does string-based neural MT learn source syntax? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1526–1534, Austin, Texas, November 2016. Association for Computational Linguistics.

[189] Lei Shu, Hu Xu, and Bing Liu. Doc: Deep open classification of text documents. In *EMNLP*, 2017.

[190] Vered Shwartz, Yoav Goldberg, and Ido Dagan. Improving hypernymy detection with an integrated path-based and distributional method. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2389–2398, Berlin, Germany, August 2016. Association for Computational Linguistics.

[191] Damien Sileo, Tim Van De Cruys, Camille Pradel, and Philippe Muller. Mining discourse markers for unsupervised sentence representation learning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3477–3486, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[192] Sameer Singh, Amarnag Subramanya, Fernando Pereira, and Andrew McCallum. Wikilinks: A large-scale cross-document coreference corpus labeled via links to Wikipedia. Technical Report UM-CS-2012-015, 2012.

[193] Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. Learning syntactic patterns for automatic hypernym discovery. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1297–1304. MIT Press, 2005.

[194] Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. Semantic taxonomy induction from heterogenous evidence. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 801–808, Sydney, Australia, July 2006. Association for Computational Linguistics.

[195] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, 2013.

[196] Yangqiu Song and Dan Roth. On dataless hierarchical text classification. In *AAAI*, 2014.

[197] Robert Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[198] Valentin I. Spitkovsky and Angel X. Chang. A cross-lingual dictionary for English Wikipedia concepts. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 3168–3175, Istanbul, Turkey, May 2012. European Languages Resources Association (ELRA).

[199] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*, 2019.

[200] Idan Szpektor and Ido Dagan. Learning entailment rules for unary templates. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 849–856, Manchester, UK, August 2008. Coling 2008 Organizing Committee.

[201] Idan Szpektor, Hristo Tanev, Ido Dagan, and Bonaventura Coppola. Scaling web-based acquisition of entailment relations. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 41–48, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[202] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[203] Duyu Tang, Bing Qin, and Ting Liu. Document modeling with gated recurrent neural network for sentiment classification. In *EMNLP*, 2015.

[204] Shuai Tang and Virginia R. de Sa. Exploiting invertible decoders for unsupervised sentence representation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4050–4060, Florence, Italy, July 2019. Association for Computational Linguistics.

[205] Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*, 2019.

[206] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. Fever: a large-scale dataset for fact extraction and verification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, 2018.

[207] Trieu H Trinh and Quoc V Le. Do language models have common sense? 2018.

[208] Trieu H Trinh and Quoc V Le. A simple method for commonsense reasoning. *arXiv preprint arXiv:1806.02847*, 2018.

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

[210] Andreas Vlachos and Sebastian Riedel. Fact checking: Task definition and dataset construction. In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*, pages 18–22, Baltimore, MD, USA, June 2014. Association for Computational Linguistics.

[211] Denny Vrandečić and Markus Krötzsch. Wikidata: A free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85, September 2014.

[212] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint arXiv:1905.00537*, 2019.

[213] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355. Association for Computational Linguistics, 2018.

[214] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium, November 2018. Association for Computational Linguistics.

[215] Alex Wang, Amapreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the Interantional Conference on Learning Representations (ICLR)*, 2019.

[216] Guoyin Wang, Chunyuan Li, Wenlin Wang, Yizhe Zhang, Dinghan Shen, Xinyuan Zhang, Ricardo Henao, and Lawrence Carin. Joint embedding of words and labels for text classification. In *ACL*, 2018.

[217] Pu Wang, Charlotta Domeniconi, and Jian Hu. Towards a universal text classifier: Transfer learning using encyclopedic knowledge. In *Data Mining Workshops*, 2009.

[218] Sida Wang and Christopher D Manning. Baselines and bigrams: Simple, good sentiment and topic classification. In *ACL*, 2012.

[219] Su Wang, Eric Holgate, Greg Durrett, and Katrin Erk. Picking apart story salads. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1455–1465. Association for Computational Linguistics, 2018.

[220] Wei Wang, Bin Bi, Ming Yan, Chen Wu, Zuyi Bao, Liwei Peng, and Luo Si. Structbert: Incorporating language structures into pre-training for deep language understanding. *arXiv preprint arXiv:1908.04577*, 2019.

[221] William Yang Wang. "liar, liar pants on fire": A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 422–426, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[222] Yizhong Wang, Sujian Li, and Jingfeng Yang. Toward fast and accurate neural discourse segmentation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 962–967, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[223] Kellie Webster, Marta Recasens, Vera Axelrod, and Jason Baldridge. Mind the GAP: A balanced corpus of gendered ambiguous pronouns. *Transactions of the Association for Computational Linguistics*, 6:605–617, 2018.

[224] Janyce Wiebe, Theresa Wilson, and Claire Cardie. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39(2):165–210, 2005.

[225] John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. Towards universal paraphrastic sentence embeddings. In *Proceedings of International Conference on Learning Representations*, 2016.

[226] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[227] Sam Wiseman, Alexander M. Rush, and Stuart M. Shieber. Learning global features for coreference resolution. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 994–1004, San Diego, California, June 2016. Association for Computational Linguistics.

[228] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771, 2019.

[229] Yadollah Yaghoobzadeh and Hinrich Schütze. Corpus-level fine-grained entity typing using contextual information. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 715–725, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[230] Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. Joint learning of the embedding of words and entities for named entity disambiguation. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 250–259, Berlin, Germany, August 2016. Association for Computational Linguistics.

[231] Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. Learning distributed representations of texts and entities from knowledge base. *Transactions of the Association for Computational Linguistics*, 5(1):397–411, 2017.

[232] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019.

[233] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *NACCL-HLT*, 2016.

[234] Liang Yao, Chengsheng Mao, and Yuan Luo. Kg-bert: Bert for knowledge graph completion. *arXiv preprint arXiv:1909.03193*, 2019.

[235] Alexander Yates and Oren Etzioni. Unsupervised methods for determining object and relation synonyms on the web. *J. Artif. Int. Res.*, 34(1):255–296, March 2009.

[236] Qingyu Yin, Yu Zhang, Weinan Zhang, Ting Liu, and William Yang Wang. Zero pronoun resolution with attention-based neural network. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 13–23, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics.

[237] Wenpeng Yin and Dan Roth. TwoWingOS: A two-wing optimization strategy for evidential claim verification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 105–114, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[238] Dani Yogatama, Chris Dyer, Chris Ling, and Phil Blunsom. Generative and discriminative text classification with recurrent neural networks. In *arXiv*, 2017.

[239] Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. Swag: A large-scale adversarial dataset for grounded commonsense inference. *arXiv preprint arXiv:1808.05326*, 2018.

[240] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics.

[241] Torsten Zesch and Iryna Gurevych. Analysis of the Wikipedia category graph for NLP applications. In *Proceedings of the Second Workshop on TextGraphs: Graph-Based Algorithms for Natural Language Processing*, pages 1–8, Rochester, NY, USA, 2007. Association for Computational Linguistics.

[242] Jingqing Zhang, Piyawat Lertvittayakumjorn, and Yike Guo. Integrating semantic knowledge to tackle zero-shot text classification. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1031–1040, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[243] Xiang Zhang, Junbo Zhao, and Yann Lecun. Character-level convolutional networks for text classification. In *NIPS*, 2015.

[244] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy, July 2019. Association for Computational Linguistics.

[245] Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53, 2017.

[246] Zhi-Min Zhou, Yu Xu, Zheng-Yu Niu, Man Lan, Jian Su, and Chew Lim Tan. Predicting discourse connectives for implicit discourse relation recognition. In *Coling 2010: Posters*, pages 1507–1514. Coling 2010 Organizing Committee, 2010.

[247] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 19–27, 2015.

APPENDIX A

TEXT CLASSIFICATION WITH WIKICAT

In this section, we will describe some extra experiments we performed on WIKICAT with non-pretrained neural models. We compare how variations in constructing WIKICAT affect the model performances on CATEVAL topical classification tasks, and how splitting categories in downstream tasks affect the prediction accuracy.

A.1 Models

Given any document d and a category c, each model defines the probability that (d, c) is a correct document-category pair by

$$p(1 \mid d, c; \theta) := \sigma((v^d)^\top U_1 v^c + b_1)$$

where $v^d, v^c \in \mathbb{R}^E$ are vector representations of (d, c) and σ is the sigmoid function. We write θ to collectively denote trainable model parameters: those used in computing v^d and v^c and also $U_1 \in \mathbb{R}^{E \times E}$, $b_1 \in \mathbb{R}$. The model is trained by negative sampling: given a document d in WIKICAT (with multiple correct categories), we sample correct categories $c_1^+ \dots c_l^+$ and k incorrect categories $c_1^- \dots c_k^-$ uniformly at random and take a gradient step on

$$\sum_{i=1}^{l} \log p(1 \mid d, c_i^+; \theta) + \sum_{i=1}^{k} \log(1 - p(1 \mid d, c_i^-; \theta))$$

Once the model is trained, we can perform weakly supervised classification by predicting the argmax over a dataset-specific set of categories.

A document $d = (d_1 \dots d_m)$ consists of m sentences d_i, whereas a category c consists of

a single word sequence. In all models we compute

$$
\begin{aligned}
v^c &= \mathbf{cat}(c) \\
v^d_i &= \mathbf{sent}(d_i, v^c) && \forall i = 1 \dots m \\
v^d &= \mathbf{doc}(v^d_1 \dots v^d_m, v^c)
\end{aligned}
$$

where each boldfaced function denotes a layer with its own set of parameters. The category embedding v^c is viewed as an optional argument in the document layers: we will omit the argument when it is not used.

We consider the three models below:

WeightAVG: Let $v^w \in \mathbb{R}^E$ denote the embedding of word type w. Define a weighted averaging operation over vectors $v_1 \dots v_m$ by $\mathrm{F}_{u,a}(v_1 \dots v_m) = \sum_{i=1}^m \alpha_i v_i$ where $\alpha_i \propto \exp(u^\top v_i + a)$ and (u, a) are learned. Our first model WeightAVG ties **cat** = **sent** and defines

$$
\begin{aligned}
\mathbf{sent}(w_1 \dots w_n) &= \mathrm{F}_{u,a}(v^{w_1} \dots v^{w_n}) \\
\mathbf{doc}(v^d_1 \dots v^d_m) &= \mathrm{F}_{u',a'}(v^d_1 \dots v^d_1)
\end{aligned}
$$

WeightLSTM: Let BiL_ϕ denote a bidirectional LSTM layer with input/output dimension E and parameter ϕ. Our second model WeightLSTM ties **cat** = **sent** and defines

$$
\begin{aligned}
\mathbf{sent}(w_1 \dots w_n) &= \mathrm{F}_{u,a}(\mathrm{BiL}_\phi(v^{w_1} \dots v^{w_n})) \\
\mathbf{doc}(v^d_1 \dots v^d_m) &= \mathrm{F}_{u',a'}(\mathrm{BiL}_\psi(v^d_1 \dots v^d_1))
\end{aligned}
$$

CatAttn: Our final model CatAttn uses the same category encoder in WeightLSTM to compute v^c. Then it uses v^c to compute attention weights over words and sentences as

follows:

$$\mathbf{sent}(w_1 \dots w_n, v^c) = \sum_{i=1}^{n} \beta_i v^{w_i}$$
$$\mathbf{doc}(v_1^d \dots v_m^d, v^c) = \sum_{i=1}^{m} \gamma_i v_i^d$$

where $\beta_i \propto \exp(v^{w_i} \cdot \tanh(U_2 v^c + b_2))$ and $\gamma_i \propto \exp(v_i^d \cdot \tanh(U_3 v^c + b_3))$ for additional parameters (U_2, U_3) and (b_2, b_3).

A.2 Experiments

A.2.1 Preprocessing and Experimental Setup

The documents and category phrases are split into sentences and tokenized using NLTK [131]. We only use the first 20 sentences per document, and each sentence is truncated to at most 30 words. Stopwords[1] are removed from documents in both training and evaluation. Numbers in the text are replaced by `<num>`. The vocabulary is set to the 50,000 most frequent lowercased words from pretrained GloVe (840B, 300 dimension) embeddings [168]. Unknown words are replaced by `<unk>`. We remove punctuation from categories. Both documents and category names are lowercased.

A.2.2 Baselines

We use several weakly supervised document classification baselines, including random and most-frequent baselines. For NYTimes, these baselines choose n categories where n is the average number of labels per instance in the test set. All the following baselines embed the document and label embeddings by the same encoder, normalize the embeddings, and return

1. We use the same stopword list as ESA, which can be found at `github.com/CogComp/cogcomp-nlp/tree/master/dataless-classifier`

the label with the highest cosine similarity with the document. We evaluate a baseline that uses fixed GloVe (840B, 300-dimension) word embeddings, computing document and label embeddings using hierarchical word averaging. We also compare to FastSent [73]. Using their code, we train 300-dimensional sentence embeddings on the Toronto Books Corpus [247].

We also evaluate ELMo [173] and BERT [47] baselines. For ELMo, both the document and label name are encoded, where we average the three ELMo layers and average over positions.

As it is zero shot learning, we do not fine-tune any models, including BERT. We use BERT as a text encoder to encode both the documents and category names. We compute BERT-base-uncased sentence and label embeddings by averaging 12 layers of [CLS] token embeddings. We call it BERT CLS AVG. We also tried averaging the last layer of BERT positions (including [CLS] and [SEP]) as document and label representations. We name this method BERT LAST AVG. We also tried averaging all positions of all layers from BERT encodings (including [CLS] and [SEP]). We name it BERT ALL AVG. Note that we use BERT differently from the traditional method of fine-tuning the concatenation of text and class labels.

Our final weakly supervised baseline is Explicit Semantic Analysis (ESA), for which we use their provided code.[2] We followed the methods of dataless classification from [25]. Instead of setting a threshold on the number of concepts as in prior work, we use all Wikipedia concepts as we find this improves ESA's performance.

We also compare to supervised results from the literature, as well as two supervised models that we train ourselves. We encode each document by WeightAVG but use GloVe embeddings as the word embeddings, then train a logistic regression classifier for each dataset using its standard training set while keeping the embeddings fixed. We call this "GloVe + LR". We also train the CatAttn model in the supervised setting, where the model is trained

2. github.com/CogComp/cogcomp-nlp/tree/master/dataless-classifier

on the binary classification task of distinguishing whether a document belongs to a class or not.

A.2.3 Primary Results

Table A.1 summarizes the results. On average, our WIKICAT trained neural models outperform the baseline weakly supervised methods across the four tasks. WEIGHTAVG is the best among our three models when we average over the four tasks, suggesting it is a simple but robust model for weakly supervised text classification.

We observe large performance gaps between our WIKICAT-trained models and other weakly supervised methods on DBPEDIA, which is unsurprising since DBPEDIA is created from Wikipedia. AGNEWS and NYTIMES contain news text, and YAHOO contains web text, both of which differ from the domain of our WIKICAT training data, but we also typically outperform most weakly supervised baselines on these datasets as well.

Among the baselines, ESA performs significantly better than the other methods on average, though ELMo performs better on AGNEWS. ELMo AVG performs better on these tasks than all BERT models tested. This is consistent with the observation from [175] that BERT is better when fine-tuned for the task of interest, while ELMo is effective even without fine-tuning.

The state-of-the-art models with supervised training data outperform our WIKICAT trained neural models. That leaves room for further improvement with weakly supervised methods.

A.2.4 Category Splitting

Sometimes a category is a combination of multiple categories. For instance, "Science & Technology" from AGNEWS can be split into two categories, "Science" and "Technology". We find it beneficial to split such cases in our weakly supervised text classification settings.

Table A.2 compares the performance of splitting vs. not splitting on AGNEWS and

	AG	DBP	YAHOO	NYT	AVG
Baselines					
Random	25.0	7.1	10.0	4.4	11.6
Most frequent	25.0	7.1	10.0	14.2	14.1
GloVe	31.6	40.3	33.6	10.9	29.1
FastSent	45.4	46.7	31.2	14.1	34.3
ESA	71.9	62.5	39.6	25.1	49.8
BERT CLS AVG	25.0	9.1	10.0	11.2	13.8
BERT LAST AVG	30.2	30.0	19.0	9.1	22.1
BERT ALL AVG	37.3	36.5	24.2	8.5	26.6
ELMo AVG	72.7	59.4	30.2	15.1	44.3
wikicat-trained models					
WeightAVG	74.2	73.4	46.1	36.7	57.6
WeightLSTM	**75.8**	74.0	45.9	22.4	54.5
CatAttn	71.2	66.7	**47.1**	32.3	54.3
Supervised models					
GloVe + LR	87.6	93.5	65.6	71.9	79.6
CatAttn	91.0	97.5	70.7	64.6	81.0
ngrams TFIDF	92.4	98.7	68.5	-	-
ULMFiT	95.0	99.2	-	-	-
DPCNN	93.1	99.1	76.1	-	-
LEAM	92.5	99.0	77.4	-	-
Human	83.8	88.2	75.0	-	-

Table A.1: Accuracy on AGNews, DBpedia, and Yahoo, and LRAP on the NYTimes dataset. TFIDF is from [243], ULMFiT is from [79], DPCNN is from [92], and LEAM is from [216]. wikicat trained weakly supervised models are based on the dataset with no category edges. The best weakly supervised performance is shown in boldface.

Yahoo. In most cases, the models benefit from classifying finer-grained categories after splitting. This is especially true for AGNews. Yahoo labels are from online forum categories so there is more noise in the ground truth labels. We find the impact of splitting category names in Yahoo to be more complex.

	AGNews		YAHOO	
	split	non-split	split	non-split
BERT ALL AVG	37.3	35.3	24.2	16.7
ELMo AVG	72.7	73.3	30.2	30.4
ESA	71.9	71.2	39.6	29.7
WeightAVG	74.2	68.8	46.1	48.8
WeightLSTM	75.8	69.1	45.9	43.2
CatAttn	71.2	65.4	47.1	42.0

Table A.2: Splitting vs. not splitting category names.

model	edges	**AG**	**DBP**	**YAHOO**	**NYT**	**AVG**
WeightAVG	0	74.2	**73.4**	46.1	36.7	**57.6**
	≤ 1	74.7	62.8	**47.8**	35.4	55.2
	≤ 2	**78.5**	61.5	39.3	**39.2**	54.6
CatAttn	0	71.2	**66.7**	**47.1**	32.3	**54.3**
	≤ 1	74.5	57.8	44.8	30.9	52.0
	≤ 2	**75.1**	61.4	42.6	**32.6**	52.9

Table A.3: Results using various numbers of edges in the Wikipedia category graph.

A.2.5 Wikipedia Category Graph Expansion in WIKICAT

In Table A.1, all WIKICAT-trained models are trained on immediate category names from the WIKICAT dataset. Since Wikipedia provides a category graph, we also experiment with training on categories that are one or two edges away from the immediate categories in the graph.

Dataset	Document	Ground Truth	Predicted
AGNews	"A Fair Tax","Some say a "fair tax" that removes the need to file tax returns from the vast majority of the citizenry is a national sales tax. This doesn't seem to be very fair to people trying to feed, house and clothe themselves and seems to ...	science technology	business
Yahoo	"Why are so many people OBSESSED with movies stars and their lives? (Why did Brad and Jenn break up?)?","Who cares!! lol Am I just too old to care about these things? I don't even KNOW these people- why would I care about what ...	business finance	society culture
Yahoo	"Do you believe in abortion?","Are you pro-life, pro-choice, or both?","Only in the cases where the abortion is absolutely called for. You know, in cases where the child is determined (pre-natally) to be severly deformed, or severely retarded, ...	politics government	society culture
DBpedia	"Ouanoukrim","Ouanoukrim (also Ouenkrim) is a mountain in Morocco located south of Marrakesh. It has two summits Timzguida (4089 m or 13415 ft) and Ras Ouanoukrim (4083 m or 13396 ft) which are the second and third highest peaks ...	nature	village
DBpedia	"Night Below","Night Below: An Underdark Campaign often known simply as Night Below is a boxed set for the second edition of the Advanced Dungeons & Dragons fantasy role-playing game. The set with the product code TSR 1125 ...	written work	company

Table A.4: Examples of errors made by the CatAttn model.

	25%	50%
KG4ZeroShot	40.2	19.7
WeightAVG	73.4	75.4

Table A.5: Comparing WeightAVG performance on unseen classes with [242] on the DBpedia dataset.

The results are shown in Table A.3. We observe that models trained on immediate categories perform better on DBpedia and Yahoo, while models trained additionally on more distant categories achieve better performance on AGNews. We hypothesize that this is due to the fact that categories in AGNews are more coarse-grained (e.g., "sports" and "business") while categories in DBpedia and Yahoo are more fine-grained. Our version of the NYTimes dataset contains 100 labels, including both coarse-grained and fine-grained labels, which is likely why we see more balanced results across the number of edges used.

A.2.6 Other Weakly Supervised Methods

Table A.5 compares our wikicat-trained WeightAVG model with the weakly supervised results of [242]. Our setting of training on the wikicat dataset and evaluating on the downstream tasks is different from their setting. However, we can still compare our results on the unseen classes[3] of the DBpedia dataset to their reported results. The percentage of seen categories affects their model performance. By leveraging the category information from wikicat, we observe a huge gain compared to their results.

3. https://github.com/JingqingZ/KG4ZeroShotText/blob/master/data/zhang15/dbpedia_csv/dbpedia_random_group_0.25.txt and https://github.com/JingqingZ/KG4ZeroShotText/blob/master/data/zhang15/dbpedia_csv/dbpedia_random_group_0.5.txt