# A Flexible Framework for Relation Extraction in Multiple Domains

by

Geeticka Chauhan

B.S. in Computer Science, Florida International University (2017)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2019



Author .....
Department of Electrical Engineering and Computer Science
May 23, 2019

Certified by .....
Peter Szolovits
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by .....
Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

# A Flexible Framework for Relation Extraction in Multiple Domains

by

Geeticka Chauhan

Submitted to the Department of Electrical Engineering and Computer Science
on May 23, 2019, in partial fulfillment of the
requirements for the degree of
Master of Science in Computer Science and Engineering

## Abstract

Relation Extraction (RE) refers to the problem of extracting semantic relationships between concepts in a given sentence, and is an important component of Natural Language Understanding (NLU). It has been popularly studied in both the general purpose as well as the medical domains, and researchers have explored the effectiveness of different neural network architectures. However, systematic comparison of methods for RE is difficult because many experiments in the field are not described precisely enough to be completely reproducible and many papers fail to report ablation studies that would highlight the relative contributions of their various combined techniques. As a result, there is a lack of consensus on techniques that will generalize to novel tasks, datasets and contexts.

This thesis introduces a unifying framework for RE known as `REflex`, applied on 3 highly used datasets (from the general, biomedical and clinical domains), with the ability to be extendable to new datasets. `REflex` allows exploration of the effect of different modeling techniques, pre-processing, training methodologies and evaluation metrics on a dataset of choice. This work performs such a systematic exploration on the 3 datasets and reveals interesting insights from pre-processing and training methodologies that often go unreported in the literature. Other insights from this exploration help in providing recommendations for future research in RE.

`REflex` has experimental as well as design goals. The experimental goals are in identification of sources of variability in results for the 3 datasets and provide the field with a strong baseline model to compare against for future improvements. The design goals are in identification of best practices for relation extraction and to be a guide for approaching new datasets.

Thesis Supervisor: Peter Szolovits
Title: Professor of Electrical Engineering and Computer Science

# Acknowledgments

With all major life events, there are many people to thank. First and foremost, I attribute my learning and accomplishments to the most important person in my life: my mother. A researcher is accomplished due to the work they perform, but true success is determined from the virtues a person preserves throughout their lifetime. I hope that I can continue doing that, as my mom taught me in my early life.

Graduate school is challenging, and I would not have been able to get thus far without my strong support system of friends, family and colleagues. My advisor, Peter Szolovits, has been incredibly supportive in my journey, and has inspired me to be a better researcher when times were tough. I could not have asked for a more supportive and encouraging advisor! I also feel very fortunate to have met my undergraduate advisor, Mark Finlayson. He was always dedicated to my success and continues to provide helpful advice even today!

It has been a pleasure to be part of the Clinical Decision-Making group and getting very helpful advice from my co-workers Matthew, Elena, Willie and Tzu-Ming (Harry). Matthew has been kind enough to brainstorm important ideas, and has taken the time to give very thoughtful feedback on much of my work.

Friends who have been constant sources of encouragement: Emellyn, Hsin-Yu, Linda, Olivia, Andrea, Sami Yamani, AJ, Andrea, Sakshi, Niharika, Preksha, Asmita and Anupam. Your presence has made my life vibrant and full of joy. Living in Sidney-Pacific is one of the best decisions I made after coming to MIT and the people I have met have inspired me to be a better version of myself. Serving as brunch chair has allowed me to take fulfilling breaks, and learn soft skills that help me in my research and everyday life. Various other student organizations have allowed me to meet dedicated MIT students who are willing to go to great lengths to fight for the causes they believe in.

I feel lucky to be a graduate student here, and continue to be inspired by the kind and hard-working people that surround me. Being able to spend everyday here is a dream beyond anything I hoped for.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

*This chapter motivates the need for a unifying framework for Relation Extraction (RE). It is also meant as an overview of current modeling approaches in the field, to show the breadth of the research performed in the past.*

## 1.1 Motivation

Relation Extraction (RE) has gained a lot of interest from the community with the introduction of the Semeval tasks from 2007 by Girju et al. [18] and 2010 by Hendrickx et al. [22]. The field is a subset of information extraction (IE) with the goal of finding semantic relationships between concepts in a given sentence, and is an important component of Natural Language Understanding (NLU). Applications include automatic knowledge base creation, question answering, as well as analysis of unstructured text data. Since the introduction of RE tasks in the general and medical domains, many researchers have explored the performance of different neural network architectures on the datasets.

However, progress in RE is hampered by reproducibility issues as well as the difficulty in assessing which techniques in the literature will generalize to novel tasks, datasets and contexts. This thesis introduces `REflex`, an open source unifying framework for RE, that allows researchers to perform various modeling and model-complementing explorations on a new dataset of their choice (see section 1.4 for goals and contribu-

tions).

This chapter is organized as follows: section 1.2 gives a brief introduction to RE, followed by two literature review sections. The first is a quantitative review section 1.3.1 providing evidence of the problems hindering progress in the field. Following this is a methods literature review section 1.3.2, which reviews past modeling techniques and existing open source frameworks and evaluation studies in NLP. The chapter concludes by summarizing the problems hindering progress in RE, and provides contributions made by the thesis to address these problems.

## 1.2 Relation Extraction Definition

Relation Extraction (RE) is a popular task in Natural Language Processing (NLP) research, and the goal of RE is to find semantic relationships between entities in a document. A relation is defined as a function $t = r(e_1, e_2, ...e_n)$ where $e_i$ are entities in a predefined relation $r$ in a document $D$. More commonly, the community considers binary relations of the form *father-of(Manuel Blum, Avrim Blum)*. Relation Classification (RC) is a subset of RE that involves distinguishing between relation types as opposed to detecting whether a relation exists between entities.

This task has been commonly applied in the general as well as biomedical domains. In particular, Ravichandran and Hovy [57] employ the use of relational patterns for answering factoid questions related to topics such as *birthdate*, *location* and *definition*. Zhang et al. [86] apply a neural model to the slot-filling task (an alias for relation classification rather than extraction), which assists in populating knowledge bases. They predict varied relations such as *spouse*, *siblings* and *title*.

In the biomedical domain, Liu et al. [44] extracted protein-protein interactions using a feature-based approach with a support vector machine (SVM) classifier. The relation they try to discover is a tertiary relation between a protein, organism and a location. In a sentence like, *Exoenzyme S is an extracellular product of Pseudomonas aeruginosa*, they predict the existence of the *Protein-Organism-Location* relation between *Exoenzyme S*, *Pseudomonas aeruginosa* and *extracellular*. In the biomedical

domain, relation extraction can have important applications such as assisting in drug discovery and in detection of cancerous genes [3]. In particular, drug-drug interaction extraction [67] is useful in allowing for automatic identification of drug interactions, in order to reduce the time spent by health care professionals in reviewing the medical literature. This detection is also an important research area in patient safety as the interactions can have life threatening effects.

## 1.3 Literature Review

This section consists of two types of literature reviews: a quantitative one providing evidence into the problems hindering progress in RE and a methods one, introducing the modeling and evaluation techniques commonly used in the field.

### 1.3.1 Quantitative Literature Review

To motivate the problems hampering progress in RE, I performed a systematic search process as of February 2019 by looking at the *cited by* list on Google Scholar (roughly ordered by number of citations) of the 3 dataset papers: Hendrickx et al. [22] (`semeval`), Segura-Bedmar et al. [67] (`ddi`) and Uzuner et al. [71] (`i2b2`). I skimmed through the first 40 papers for the `semeval` paper, 110 papers for the `ddi` paper and 578 papers for the `i2b2` paper, looking specifically for neural relation extraction papers.

Upon applying this filtering procedure, I found 22 papers for `semeval` (+ 4 papers that were not in the search list, but were cited in section 1.3.2), 15 papers for `ddi` (+ 2 papers from the section 1.3.2) and 12 papers for `i2b2`. There was an overlap of 2 papers in the `semeval` and `ddi` list, but since they were being applied to the biomedical tasks, I decided to move them to the `ddi` list. Finally, there were 24 papers for `semeval`, 17 papers for `ddi` and 12 papers for `i2b2`. For the final list of papers, please refer to Appendix A. In total, there are 53 relevant neural RE papers discussed in the following subsections, filtered from a total of 728 papers.

### Reproducibility

Reproducibility is important for validating previous work and building upon it [16]. Lack of reproducibility can be attributed to many factors such as difficulty in availability of source code [24] and omission of sources of variability such as hyperparameter details [12].

Only 16 out of the 53 relevant papers had released their source code. In the `semeval` list, only 6 out of 24 total papers had source code available. This number was 6 out of 17 for `ddi` and 4 out of 12 for `i2b2`. Additionally, much of this code was lacking in modularity to be easily extendable to new datasets. In many cases, the process of reproducing the paper results was also unclear and lack of proper documentation made this more difficult.

Models were more frequently evaluated on only one dataset. However, papers in the general domain often evaluated their models on a larger number of datasets than the biomedical domain. In `semeval`, an average of 1.75 datasets were evaluated, with 8 papers being evaluated on more than 1 dataset. Out of these papers, one was evaluated on 6 datasets and the other was evaluated on 7 datasets. Only one of these papers had source code available, which was not mentioned in the paper and was found by additional search on Google. In `ddi`, 1.23 datasets were evaluated on average with 4 papers being evaluated on 2 datasets. For `i2b2`, 1.42 datasets were evaluated on average, and this number was driven by one paper evaluated on 5 datasets whose source code was not publicly available.

Most papers from the list mentioned some hyperparameter details. However, the list was often incomplete, and the common missing hyperparameters were *number of epochs*, *batch size* and whether a random initialization seed was set for the model or the random functions used in the code. Some papers that used the early stop mechanism were missing information about the size and criterion of the early stop evaluation data. Papers also failed to mention if a specific hyperparameter search strategy like grid search, manual search, or random search was performed [5].

#### Ablation Studies

Ablation studies are important in understanding the sources of variation in results as well as which parts of the model drive performance. While 20 of the 24 papers in the `semeval` list performed ablation studies, very few from the `ddi` and `i2b2` list performed them. 7 of 17 papers performed an ablation study in `ddi` and 3 of 12 papers did so for `i2b2`. In ablation studies and other reported experiments, key details related to pre-processing were missing, which we found critical in our experiments.

### 1.3.2 Methods Literature Review

#### Survey of Modeling Techniques

Given the popularity of neural relation extraction in the recent years, there is an abundance of papers that apply similar techniques to different datasets. Despite neural relation extraction existing since 2012, the biomedical domain saw a less rapid application of these models as compared to the general domain, as seen in the following subsections. And even though these papers investigated different neural network architectures for this task, no studies were published that explored the extent of improvement offered by non-modeling techniques such as pre-processing, evaluation techniques and hyperparameter tuning techniques for RE.

**General domain** Relation extraction over the general purpose domain has seen rapid progress in recent years with the introduction of the SemEval 2007 and 2010 tasks on relation classification between pairs of nominals, as well as 2018 task on relation extraction and classification in scientific papers [18, 22, 17]. The submissions to the 2007 and 2010 tasks involved the use of varied classification models such as Naive Bayes, k-nearest neighbor (k-NN), Maximum Entropy (MaxEnt) and SVM classifiers. Neural Network (NN) applications to NLP were only made popular in 2011 by Collobert et al.. In 2012, Socher et al. [69] applied a matrix-vector based recursive neural network (MVRNN) using a syntactic parse tree feature to improve performance for the SemEval

2010 dataset on top of existing non-neural techniques. However, Zeng et al. [81] were the first to applied a model not based on semantic features. They applied a Convolutional Neural Network (CNN) architecture with novel position-based features to the same task and achieve a better performance than MVRNN. Since then, the field of neural relation extraction saw many advances. In 2015, Zeng et al. [82] introduced a distant supervision technique for relation classification using a multi-pooling approach over CNNs. In the same year, various other CNN based approaches were introduced [66] (`CRCNN` model used in my experiments) and [76] and so were Recurrent Neural Network (RNN) based approaches [84, 15, 77]. 2016 saw even more complex models and better performance on relation classification [50, 73, 8, 78]. Finally, more recent methods explored different architectures beyond the standard RNN and CNN, by using graph convolutions over dependency trees of the sentences [88]. Another recent method reduced relation extraction to answering simple reading comprehension questions [34].

**Biomedical Domain** Advances in the biomedical domain have been inspired from techniques in the general domain, but have happened at a slower pace. There exist relation extraction challenges in this domain as well, including the drug-drug interaction extraction task known as DDI Extraction [67] (`ddi`) and the relation classification task on clinical notes [71]. For both challenges, participants submitted non-neural models, but there has been considerable work on these datasets since their respective years. Despite the many modeling techniques proposed by researchers working in the general purpose domain, most papers built on top of the idea of using CNN with position-based embeddings from [81]. Even for more recent tasks involving relation extraction from scientific abstracts [2, 17], this modeling technique seems to be a common baseline [32, 62, 26]. The first time a neural model was applied to `ddi` was in 2016 by Liu et al. [43]. The model involved dataset specific pre-processing on top of the CNN with the position features model proposed by Zeng et al.. Around the same time and with similar performance, Zhao et al. [89] introduced a syntax

CNN method, that made use of word embeddings based on the syntactic parse of the sentence on top of position embeddings as well as grammatical features. In the same vein as the multi-pooling approach by Zeng et al. [82], Luo et al. [47] proposed a segmented CNN approach with position embeddings in 2017 by dividing the sentence into 5 parts based on the position of the entities. Similarly, He et al. [21] applied a multi-pooling architecture on top of a CNN with position embeddings and a loss function with a category-level constraint matrix. The same authors also explored a unified CNN-RNN architecture in [20]. Similar to the shortest dependency path idea by Xu et al. [77], Li et al. [35] used an RNN along the shortest dependency path along with character-based convolutions to extract relations in two common biomedical datasets. Finally, a recent discussing the improvement that character embeddings can provide for a biomedical dataset is Nguyen and Verspoor [51]. Character embeddings is a popular idea employed previously in the relation extraction domain [32].

### Prior frameworks and studies

**Existing open source frameworks** Literature review suggests that the field of relation extraction would benefit from an open source, extendable and transparent framework. While there do exist recently created frameworks for RE such as Björne and Salakoski [7] and Kang et al. [29], they are based on a support vector machine (SVM). There does not exist a generalizable neural network-based framework for this field. In terms of existing products, Amazon released the Amazon Comprehend Medical API, allowing relation extraction for clinical notes, but this is more of a black-box model, which is not as beneficial to the research community.

**Existing evaluation studies** Even though there is a gap between the general and medical relation extraction domains at the moment, more mainstream research is now being applied to medical datasets. A recent study by Mandya et al. [49] employed a combined LSTM-CNN model for cross-sentence relation

extraction to `semeval` and a biomedical dataset with the aim to show state of the art performance on the biomedical domain. Additionally Zhao et al. [89] provided a detailed ablation study on the effect of performance provided by negative instance filtering, which is a pre-processing technique specific to `ddi`, as well as the modeling techniques that they choose. Outside the relation extraction domain, impact of non-modeling techniques is being studied more recently, with Reimers and Gurevych [58] reporting the effect of different hyperparameters in the performance for the named entity recognition (NER) task. The same authors also studied the effects of random initialization seeds to the performance of models in [60], with the conclusion that comparing score distributions of two models is much more impactful than simply comparing one evaluation score. Recently, [14] discussed similar problems for the question-answering field. The effects of pre-processing for sentiment analysis and text categorization were tested in [9]. Addressing the replication and reproduction issue for NER and Wordnet:Similarity tasks is an older work by Fokkens et al. [16]. They spoke about the impact of different non-modeling techniques such as preprocessing, experimental setup, versioning, system output and system variation for these tasks and conclude that these categories are important to explore in order to maintain reproducibility of results. Another recent paper aiming to understand the text processing capabilities of CNN filters is Jacovi et al. [25]. RE would benefit from such studies to understand the true source of performance gains in results. Current studies in RE are local in nature in that they simply focus on the improvement offered by modeling techniques rather than those provided by non-modeling techniques, such as a range of pre-processing techniques.

## 1.4 Goals and Contributions

Given the lack of detailed evaluation studies in RE, it is difficult to assess the causes of large variability of results, which makes a *fair comparison* of models a difficult task. An open-source unifying framework enabling the comparison of various training methodologies, pre-processing, modeling techniques and evaluation metrics would help add clarity to what techniques add true performance and generalize best. The contributions of this work is as follows:

1. An open-source unifying framework known as `REflex`[1], that is extendable to new datasets.

2. Exploration of modeling and model-complementing (training methodologies and pre-processing) techniques on 3 popular RE datasets, along with a discussion of the implications of different evaluation metrics, particularly for the medical settings.

[1] code available at https://github.com/geetickachauhan/relation-extraction

# Chapter 2

# Data

*This chapter introduces the different datasets used for the exploratory piece of the thesis, mentioning intricacies such as their class distributions and inter-annotator agreements. These details will be critical in latter chapters when I discuss the appropriateness of the evaluation metrics for new as well as the current datasets. A summary of important information about these datasets is also present in table 2.4. For the rest of the thesis, medical datasets refer to the* `ddi` *and* `i2b2` *datasets. All datasets were evaluated for the classification task, but the medical datasets were also evaluated for the detection task (explained more in section 3.1.4 of chapter 3).*

## 2.1 Semeval 2010

`semeval` [22] consists of 8000 training sentences as well as 2,717 test sentences for the multi-way classification of semantic relations between pairs of nominals. There are a total of 19 relations (where 18 relations consist of taking directionality into account), with an *Other* class which is considered noisy, with annotators classifying this class if no fit was found in the other classes. The official evaluation reported macro-F1 scores and did not count the *Other* class in calculations. Inter-annotator agreement for this dataset is between *60%* and *95%*. The class distribution in the Semeval 2010 dataset is listed in table 2.1.

| Class Name | Num | Example |
|---|---|---|
| Cause-Effect(e1,e2) | 688 | those cancers were caused by radiation exposures |
| Cause-Effect(e2,e1) | 1318 | |
| Component-Whole(e1,e2) | 940 | my apartment has a large kitchen |
| Component-Whole(e2,e1) | 942 | |
| Content-Container(e1,e2) | 748 | a bottle full of honey was weighed |
| Content-Container(e2,e1) | 332 | |
| Entity-Destination(e1,e2) | 1688 | the boy went to bed |
| Entity-Destination(e2,e1) | 2 | |
| Entity-Origin(e1,e2) | 1136 | letters from foreign countries |
| Entity-Origin(e2,e1) | 296 | |
| Instrument-Agency(e1,e2) | 194 | phone operator |
| Instrument-Agency(e2,e1) | 814 | |
| Member-Collection(e1,e2) | 156 | there are many trees in the forest |
| Member-Collection(e2,e1) | 1224 | |
| Message-Topic(e1,e2) | 980 | the lecture was about semantics |
| Message-Topic(e2,e1) | 288 | |
| Product-Producer(e1,e2) | 646 | a factory manufactures suits |
| Product-Producer(e2,e1) | 788 | |
| Other | 2820 | - |

Table 2.1: Class distribution of `semeval`. The column Num = number of examples per relation. Because the relation names are self-explanatory, examples of relation types are provided with the involved entities underlined. *Other* class omits the example column because it is noisy and does not have a representative example.

## 2.2 DDI Extraction

`ddi` [67] consists of 1,017 texts with 18,491 pharmacological substances and 5,021 drug-drug interactions from PubMed articles in the pharmacological literature. A total of 5 relations are present with a *None* class indicating no interaction between the drug pairs. The official evaluation reported macro-F1 scores for classification, along with a detection macro-F1. While classification was a multi-class classification task, detection converted the problem into a binary classification between non-*None* classes and *None* classes (explained more in chapter 3). The class distribution is listed in table 2.2.

| Class Name | Num | Description |
|---|---|---|
| Advise | 863 | recommendation or advice related to the use of the two drugs |
| Effect | 1591 | effect related to the drug-drug interaction |
| Int | 228 | an interaction occurs without providing more information |
| Mechanism | 1299 | mechanism of the interaction |
| None | 21948 | No interaction |

Table 2.2: Class distribution of `ddi`. The column Num = Number of examples per relation.

## 2.3 i2b2/VA 2010 relations

`i2b2` [71] consists of discharge summaries from Partners Healthcare and the MIMIC II Database [63]. They released 394 training reports, 477 test reports and 877 unannotated reports for this purpose. After the challenge, only a part of the data was publicly released for research and the dataset consists of 8 non-*None* relations in three categories: Medical *Problem* - *Problem*, *Problem* - *Test* and *Problem* - *Treatment* relations. There were also *None* relations present in each of the three categories. The official evaluation reported micro-F1 scores and did not count the *None* class in calculations. The class distribution is listed in table 2.3. The table lists the 3 different types of *None* classes, but the model was fed in all of these as the *None* type. This does not make a difference in score reporting because these classes were not included in official evaluation scores for this dataset.

| Class Name | Num | Description |
|---|---|---|
| PIP | 2203 | Medical problem indicates medical problem |
| PP-None | 12506 | - |
| TeCP | 504 | Test conducted to investigate medical problem |
| TeRP | 3053 | Test reveals medical problem |
| TeP-None | 2964 | - |
| TrAP | 2617 | Treatment is administered for medical problem |
| TrCP | 526 | Treatment causes medical problem |
| TrIP | 203 | Treatment improves medical problem |
| TrNAP | 174 | Treatment is not administered because of medical problem |
| TrWP | 133 | Treatment worsens medical problem |
| TrP-None | 4462 | - |

Table 2.3: Class distribution of `i2b2`. The column Num = number of examples per relation.

| Dataset | Relations | Evaluation Metric | Interannotator Agreement | Detection |
|---|---|---|---|---|
| semeval | 18 | Macro-F1 | 0.6-0.95 | No |
| ddi | 5 | Macro-F1 | >0.8; 0.55-0.72 | Yes |
| i2b2 | 8 | Micro-F1 | - | Yes |

Table 2.4: Dataset information, with column Detection referring to whether detection task from section 3.1.4 in chapter 3 was evaluated on. Relations column only includes relations used in the official evaluation metric. ddi was built from two separately annotated sources and therefore contains two interannotator agreements.

## 2.4 Official Challenge Tasks Compared Against

In this work, I compare my results against the following challenge tasks for each dataset:

- Semeval 2010 task 8: Multi-Way Classification of Semantic Relations between Pairs of Nominals
- DDI Extraction 2013 task 9.2: Extraction of drug-drug interactions
- i2b2/VA 2010 challenge on relation classification

# Chapter 3

# Methodology

*This chapter introduces the conceptual picture of the framework, briefly mentioning the higher level details of the stages in the pipeline. Background information related to the baseline model used and evaluation metrics is mentioned in the latter sections.*

## 3.1 Framework

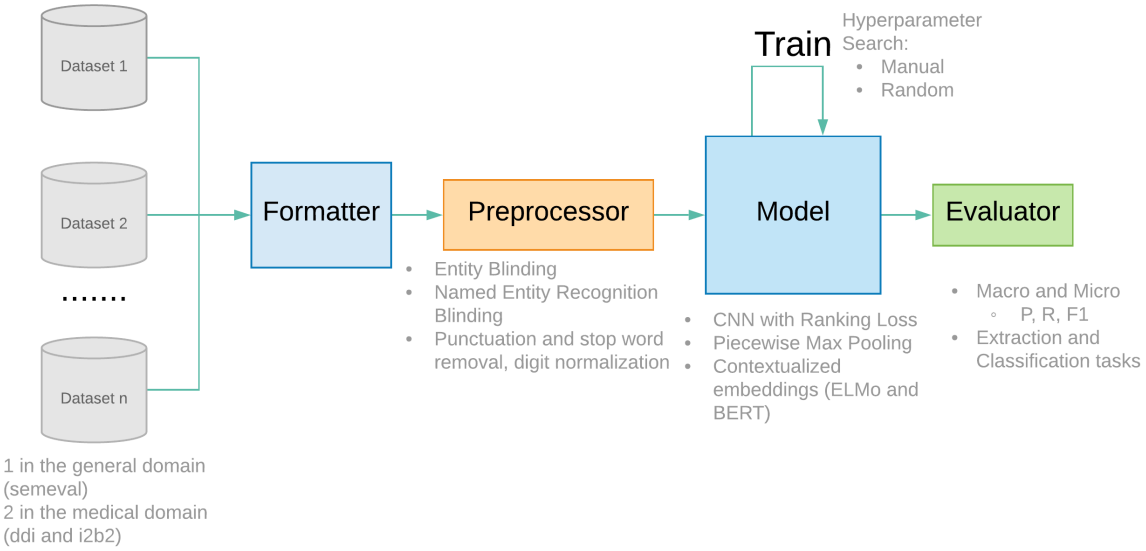


Figure 3-1: Systematic exploration framework. Each dataset results computed separately.

The framework breaks up various parts of processing into different stages, allowing for modular addition of components in the future. First, a `formatter` converts the raw dataset into a common input format accepted by the `preprocessor`, and the pre-processed dataset is then fed to the `model`. The `model` then performs the training after which `evaluator` performs evaluation on the test set (or development set for cross validation). With this framework, I run the experiments described in the following subsections:

### 3.1.1 Pre-Processing

Various pre-processing methods are tested after performing simple tokenization and lower-casing of the words: entity blinding used by Liu et al. [43], commonly applied **stop-word and punctuation removal**, digit normalization applied for `ddi` in [89], and named entity recognition related replacement (this is known as NER blinding in this work). I used the spaCy framework[1] to perform tokenization as well as to identify punctuations and digits.

Stop word removal is a common technique in Natural Language Processing (NLP) to remove commonly used words such as *the* and *is* in order to simplify the sentence. The technique was first coined in Luhn [45], and was commonly used in Information Retrieval (IR) to make the processing of natural language queries faster and more accurate. This was commonly used in Information Retrieval (IR), first coined in [45].

**Digit normalization** refers to the replacement of all decimals and integers in the sentence by the word *number*. Instead of using regular expressions to search for decimals and digits, I used spaCy's `like_num` argument which identifies decimals and digits as well as language specific words like *ten* or *hundred*.

**Entity blinding** and **NER blinding** are similar concept blinding techniques where the first is performed based on gold standard annotations, while the second is performed by running NER on the original sentence. I replace the words in the sentence matching the entity or named entity span with the target label and use those for training and testing.

[1] https://github.com/explosion/spaCy

Entity labels for `semeval` were not annotated with type information, whereas `ddi` identified drugs and `i2b2` identified medical problems, tests and treatments. Therefore, entity labels for `semeval` were *ENTITY*, for `ddi` were *DRUG* and for `i2b2` were *PROBLEM*, *TREATMENT* and *TEST*. In this work, I use *fine-grained concept type* to refer to the presence of more than one concept type, as in the the case of `i2b2`.

NER labels for `semeval` consisted of those provided by the large english model by spaCy and provided standard types such as *PERSON* and *ORGANIZATION*, whereas those for the medical datasets was provided by the scispacy medium size model[2] and did not provide types. In this case, blinding consisted of replacing the words in the sentence by *Entity*.

As an example of blinding, consider the sentence in figure B-1 in chapter 4. The result of entity blinding that sentence is shown in figure 3-2.

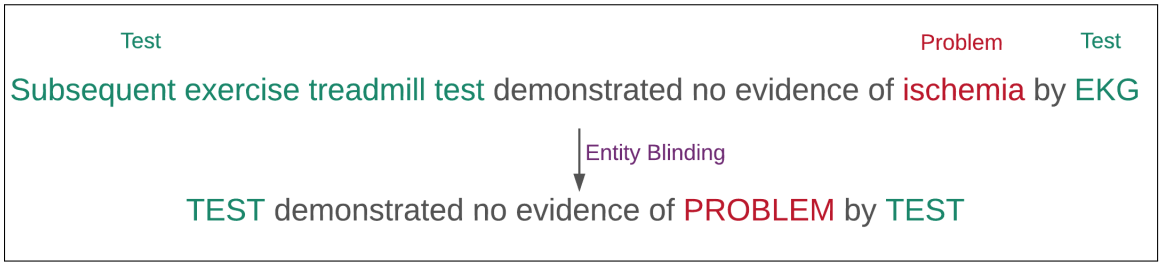


Figure 3-2: Result of entity blinding for a sentence in the `i2b2` dataset

### 3.1.2 Model

I employ a baseline model based upon [81] and [66], which is a convolutional neural network (CNN) with position embeddings and a ranking loss (referred to as `CRCNN` in this work and explained in subsection 3.2.1). The model is initialized with pre-trained word embeddings. The model for the general domain dataset is initialized with senna embeddings by Collobert et al. [13], whereas that for the medical domain (biomedical and clinical) is initialized with the `PubMed-PMC-wikipedia` embeddings released by Pyssalo et al. [54]. Many perturbations on top of `CRCNN` model are tested, such as

[2]https://allenai.github.io/scispacy/

**piecewise max-pooling**, as suggested by Zeng et al. [82] and **ELMo embeddings** Peters et al. [53]. To compare different featurizations of contextualized embeddings, I also employ the embeddings generated by the **BERT** model (rather than the standard fine-tuning approach).

The fine-tuning approach, which tends to be computationally expensive, has been thoroughly explored for multiple tasks, including medical relation extraction by Lee et al. [33], but the approach of using contextualized embeddings has not been explored in the literature as much. I chose to explore different ways of incorporating the BERT contextualized embeddings for researchers that wanted to utilize a less computationally intensive technique, while still aiming for performance gains for their task.

Because ELMo provides token level embeddings, they were concatenated with the word and position embeddings from `CRCNN` before the convolution phase. According to the terminology used in section 3.2.1, new feature embeddings were generated by concatenating the word embeddings, word position embeddings as well as the ELMo embeddings on a word-by-word basis.

BERT, in contrast, provides word-piece level as well as sentence level embeddings. The word-piece level embeddings were concatenated similar to ELMo (known as BERT-tokens) after the individual word pieces were averaged to form one word embedding. For example, if BERT split the word "playing" to generate embeddings for "play" and "##ing," I averaged the embeddings for the two word pieces to form one word embedding for "playing." The sentence level embeddings were concatenated with the fixed size sentence representation, known as $r_x$ in section 3.2.1, which is output after convolution of word and position embeddings (known as BERT-CLS).

### 3.1.3 Training Methodologies

Two ways of doing hyperparameter tuning were explored: **manual tuning** and **random search** [5].

Evaluating on 3 datasets meant that I needed to identify a default list of hyperparameters by tuning on one of the datasets before identification of the hyperparameter list for the other two. I chose `semeval` for initial tuning due to its larger literature

| Hyperparameter | Values |
|---|---|
| epoch | {50,100,150,200} |
| lr decay | [1e-3, 1e-4, 1e-5] |
| sgd momentum | {T, F} |
| early stop | {T, F} |
| pos embed | {10, 50, 80, 100} |
| filter dimension | {50, 150} |
| filter size | 2-3-4, 3-4-5 |
| batch size | {70, 30} |

Table 3.1: Hyperparameters explored for the first pass of manual search. lr decay means learning rate decay at [60, 120] epochs, pos embed refers to the position embedding size.

| Hyperparameter | Distributions |
|---|---|
| epoch | uniform(70, 300) |
| lr | {constant, decay} |
| lr init | uniform(1e-5, 0.001) |
| filter size | 2-3, 2-3-4, 2-3-4-5<br>3-4-5, 3-4-5-6 |
| early stop | {T, F} |
| batch size | uniform(30, 70) |

Table 3.2: Hyperparameter distributions for random search. Those written in {} are picked with equal probabilities. The learning rate (lr) was uniformly initialized, and decayed from 0.001 to the lr init value (used as a post decay value in this scenario) at half of the number of epochs. If early stop was true, patience was set to a fifth of the number of epochs. I ran 100-120 experiments for each dataset to search for optimal hyperparameters.

and because the `CRCNN` model was originally evaluated on this dataset. I started with reference hyperparameters listed in Zeng et al. [81] and Santos et al. [66] and identified default hyperparameters after tuning on a dev set randomly sampled from the training data of the `semeval` dataset. These default hyperparameters[3] were used as starting points for manual tuning on the medical datasets as well as random search for all datasets.

I perform manual tuning on a subset of the hyperparameters, mentioned in table 3.1. In order to avoid overfitting in cross validation pointed out by Cawley and Talbot [10], I perform a nested cross validation procedure, keeping a dev fold for hyperparameter tuning and a held out fold for score reporting.

[3]listed in source code

On these dev folds, I perform paired t-tests for each of the perturbations to the parameters listed in table 3.1. The first pass involves changing one hyperparameter per experiment and noting the ones that cause a statistically significant improvement, which helps in identification of a narrower list of hyperparameters to tune on. I further refine the hyperparameter values in our second pass by testing on values similar to those that were leading to statistically significant improvements in the first pass. For example, if I noticed that lower epoch values were helpful in the first pass, I tested them in combination with the other optimal hyperparameter values (from first pass) in the second pass.

For each of the datasets, I tune based on their official challenge evaluation metrics listed in chapter 2. `ddi` and `i2b2` had 5-fold nested cross validation performed on them, whereas `semeval` had 10-fold cross validation performed.

Random search was performed based on the official evaluation metrics for each dataset, on a fixed dev set randomly sampled from the training data. Distributions used for the search are listed in table 3.2.

### 3.1.4 Evaluation

The official challenge problems for all datasets compared models based on multi-class classification, but for the medical datasets, I was also interested in the changes in model performance if the task was treated as a binary classification problem. This was based on the rationale that in the drug literature, for example, pharmacologists would not want to sacrifice the ability to identify a potentially life threatening drug interaction pair, even if the type of the drug pair is not known. Therefore, I report results for the multi-class as well as binary classification scenario. For clarity, let us refer to them in the rest of the thesis as *classification* and *detection* respectively.

*Detection* results were obtained using our evaluation scripts by treating existing relations as one class, ignoring the types outputted by the model. The other class in this task was the *None* or *Other* class, representing non-existing relations. Note that I did not re-train the model for this task.

In addition to evaluating on 2 tasks for the medical and 1 task for the general

dataset, I comment on the implications of different evaluation metrics in section 4.2.5 of chapter 4. For example, it is important to note recall versus precision performance for a drug pair interaction setting where it is more critical to identify a potentially dangerous drug interaction, even if there is a tradeoff with precision.

## 3.2 Baseline model and Evaluation Metrics

This section introduces the CNN model architecture (`CRCNN`) by Santos et al. [66] as well as from Jin et al. [26]. It also details the evaluation metrics used in this work.

### 3.2.1 CNN Model Details

Figure 3-3 from Jin et al. [26] presents the architecture of the CNN model. The model first takes the tokenized sentence, as well as the targeted entities, and transforms it to a sequence of continuous embedding vectors (subsection 3.2.1). Next, the model uses a convolution layer to transform the embedded sentence to a fixed-size representation of the whole sentence. Finally, it computes the score for each relation class via a linear transformation (subsection 3.2.1). The overall system is trained end-to-end via a cross entropy loss augmented with a variant of negative sampling (subsection 3.2.1).

#### Feature Embeddings

Given a sentence $\boldsymbol{x} = \Big[x_1, \ldots, x_n\Big]$, the tokens $x_i$ are featurized into continuous embedding vectors via concatenated word embeddings ($e^{w_i}$) and word position embeddings ($e^{wp_i}$): $e_i = [e^{w_i}, e^{wp_i}]$. For conceptual simplicity, I refer to $\boldsymbol{x}$ as a vector instead of a matrix by collapsing the embedding level dimensionality.

**Word Embeddings** Word representations are encoded by the column vector in the embedding matrix $W^{word} \in \mathbb{R}^{d^w \times |V|}$, where $V$ is the vocabulary of the dataset. Each column $W_i^{word} \in \mathbb{R}^{d^w}$ is the word embedding vector for the $i^{\text{th}}$ word in the vocabulary. This matrix is trainable during the optimization process and initialized by pre-trained embedding vectors described in subsection 3.1.2.

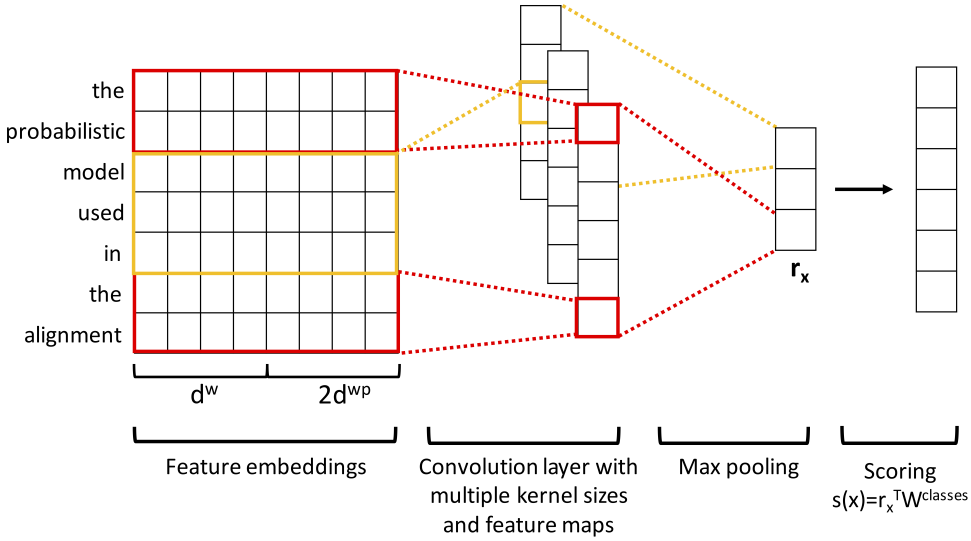


Figure 3-3: Illustration of CNN model architecture.

**Word Position Embeddings (WPEs)** In general, the information needed to determine the sentence's relations mostly comes from the words close to the two entities. In addition, some information needs to be input into the model to indicate which words are entities. The word's relative position to either entity is used as a feature to fulfill the above-mentioned two functions. For instance, in the sentence "the **probabilistic model** used in the **alignment**" shown in Figure 3-3, the relative distance of all the words to the left entity "probabilistic model" is $-1, 0, 0, 1, 2, 3, 4$ and that to the right entity "alignment" is $-6, -5, -4, -3, -2, -1, 0$. Each relative distance is mapped into a vector of dimension $d^{wp}$, which is randomly initialized then updated during training. Each word $w$ has two relative distances $wp_1$ and $wp_2$ with respect to two entities $entity_1$ and $entity_2$, and each distance is mapped to corresponding embedding vector and the position embedding $e^{wp}$ of word $w$ is the concatenation of these two vectors: $e^{wp} = [e^{wp_1}, e^{wp_2}]$.

### Sentence Representation and Scoring

After featurization, a sentence $\boldsymbol{x}$ of length $N$ is represented as $\boldsymbol{e} = [e_1, e_2, ...e_N]$ (after collapsing the embedding level dimensionality). I denote $\boldsymbol{e}_{i:i+j}$ as the concatenation

of featurized tokens: $\boldsymbol{e}_{i:i+j} = [e_i, e_{i+1}, ..., e_{i+j}]$. A convolution operation involves a filter weight matrix $W \in \mathbb{R}^{(d^w+2d^{wp})\times k}$, which is applied to a window of k words to produce a new feature $c_i$, as represented by:

$$\boldsymbol{c}_i = \tanh(W \cdot \boldsymbol{e}_{i:i+k-1} + \boldsymbol{b}),$$

where $b \in \mathbb{R}^{(d^w+2d^{wp})\times 1}$ is a bias term. This filter is applied to each possible window of words in the sentence $\boldsymbol{e}_{1:h}, \boldsymbol{e}_{2:h+1}, ..., \boldsymbol{e}_{N-h+1:N}$ to produce a feature map matrix $C = [\boldsymbol{c}_1, \boldsymbol{c}_2, ..., \boldsymbol{c}_{N-k+1}]$. For conceptual simplicity, let us collapse the matrix $C$ to a vector $\boldsymbol{c}$ by ignoring the feature embeddings dimension, $d^w + 2d^{wp}$. A max pooling operation is then applied to this feature map to obtain the maximum value $\hat{c} = max\{\mathbf{c}\}$ as the feature corresponding to this particular filter. This is how one feature is extracted by one filter. And the model can use multiple filters with varying window sizes and filter parameters to produce multiple features. All obtained features are then concatenated to form the fixed size sentence representation $r_{\boldsymbol{x}}$. Given the vector representation $r_{\boldsymbol{x}}$ of the sentence $\boldsymbol{x}$, class scores are computed via a linear transformation mediated by a trainable matrix $W^{classes}$. The relation class is then inferred by taking the index of the maximum score in the class scores, $s(\boldsymbol{x})$.

**Loss with Negative Sampling**

After obtaining the score vector $s(\boldsymbol{x})$ for the sentence $\boldsymbol{x}$, a loss function is applied, motivated by ideas in negative sampling as follows. Let $y$ be the correct label for sentence $\boldsymbol{x}$, and $I = \mathcal{Y} \setminus \{y\}$ be the set of all incorrect labels for $\boldsymbol{x}$. Then, the loss is computed:

$$L = \log\left(1 + e^{\gamma\left(m^+ - s(\boldsymbol{x})_y\right)}\right) + \log\left(1 + e^{\gamma\left(m^- + \max_{y' \in I}\left(s(\boldsymbol{x})_{y'}\right)\right)}\right),$$

where $m^+$ and $m^-$ are margins, $\gamma$ is the penalty scale factor. Minimizing this loss function will both increase the score of the correct label and decrease that of the

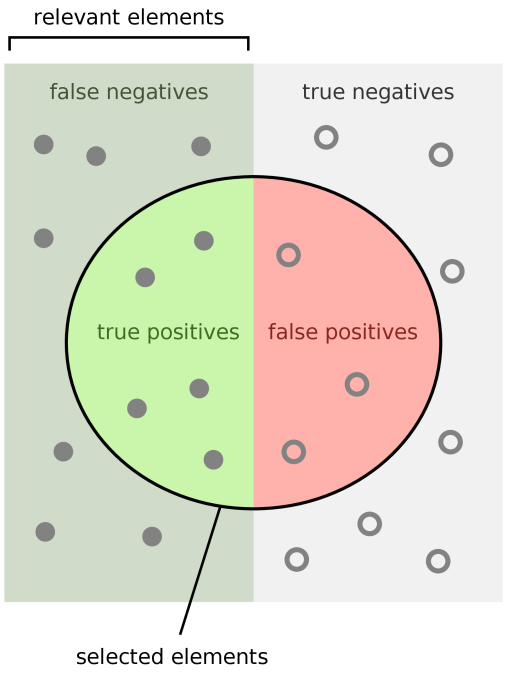


Figure 3-4: Sample space of Predictions

wrong label. Intuition behind the ranking loss function is provided in Santos et al. [66]. Adam optimizer [31] was used to minimize the loss function.

### 3.2.2 Evaluation Metrics

A helpful visualization of the sample space of predictions is provided by the figure 3-4 by Walber, licensed under a *Creative Commons Attribution-ShareAlike 4.0 International license*. Let us shorten False Positives to FP, False Negatives to FN, True Positives to TP and True Negatives to TN. In a multi-class setting, the following (except Accuracy) are calculated for each class and aggregated to compute either macro or micro statistics:

#### Precision (P)

This is a measure of the number of relevant items from those selected, also referred to as *specificity* in the literature. This is computed as $\frac{TP}{TP+FP}$.

### Recall (R)

This is a measure of the number of relevant items that are selected, also referred to as *sensitivity*. This is computed as $\frac{TP}{TP+FN}$.

### F1

A harmonic mean of Precision and Recall, it is meant to capture the performance of a model on both statistics equally. This is computed as $\frac{2*(P*R)}{P+R}$.

### Accuracy

In this work, accuracy is not calculated on a per-class basis, and is simply provided as one overall score for the model. Accuracy is a measure of the number of correct predictions out of the total predictions, and in a multi-class setting with classes $\in \{1, .., n\}$, can be computed as $\frac{TP_1+...+TP_n}{TP_1+...+TP_n+FP_1+...+FP_n}$.

### Macro vs Micro Statistics

P, R and F1 can be computed as one score as either *macro* or *micro* statistics. While *macro* statistics average the individual scores for all classes, *micro* statistics consider the individual example numbers in their calculations.

When the classes $\in \{1, ..., n\}$, macro-P will be computed as $\frac{P_1+...+P_n}{n}$. This is similar to the calculations for R and F1. In contrast, micro-P will be computed as $\frac{TP_1+...+TP_n}{TP_1+...+TP_n+FP_1+...+FP_n}$; micro-R will be computed as $\frac{TP_1+...+TP_n}{TP_1+...+TP_n+FN_1+...+FN_n}$; and micro-F1 will be computed as the harmonic mean of these calculated micro-P and micro-R.

Each metric is affected differently by class distributions, and their implications are discussed in chapter 4, subsection 4.2.5.

# Chapter 4

# Results and Discussion

*This chapter introduces the results from different experiments, and discusses important takeaways in the latter sections.*

## 4.1 Introduction

In the following section, I present the results for experiments pertaining to pre-processing, modeling and training methodologies, as described in section 3.1 of chapter 3. Appendix D reports the different evaluation metrics for each of these experiments.

Experiments on the medical datasets involved hyperparameters found after performing manual search individually on them, while those on `semeval` involved default hyperparameters. Refer to section 3.1.3 of chapter 3 for details on the experimental set up.

Using the fixed set of hyperparameters for each dataset, I tested the perturbations for pre-processing and modeling listed in tables 4.1 and 4.2. Perturbations on the hyperparameter search are listed in table 4.3 and compare performance with different hyperparameter values found using different tuning strategies.

Evaluation was performed on the standard *classification* and additional *detection* task scores, described in section 3.1.4 of chapter 3. In tables 4.1 and 4.2, these scores are reported under the *Class* and *Detect* columns respectively.

In each result table in this chapter, test results are mentioned at the top, followed

by cross validated results (with standard deviation) mentioned below in smaller font. Official test set results are reported to compare against current literature, but cross validation scores are used to perform statistical significance in the form of a paired t-test. The paired t-tests are performed on the results found from the held-out sets in all the cross validation folds. In the following discussion, performance is claimed to be worse or better only if the test set results reflect these and statistical significance is found on the held out cross validation sets.

## 4.2 Discussion

Recently, CNNs have achieved strong performance for text classification and are typically more efficient than recurrent architectures [4, 28, 74, 85]. The speed of the baseline `CRCNN` model allows the exploration of multiple alternatives for every stage of the pipeline. I discuss these results pertaining to the *classification* task for all datasets and the *detection* task for the medical datasets.

The results in this section demonstrate that 1) pre-processing choices can cause the largest variation in performance, 2) reporting scores on one test-set split is problematic due to split bias, 3) featurization technique matters for contextualized embeddings, 4) picking the right hyperparameters is important to performance and 5) picking the right evaluation metrics should be driven by class imbalance issues.

### 4.2.1 Pre-processing

Often, papers fail to mention the importance of pre-processing in performance improvements. Experiments in table 4.1 reveal that they can cause larger variations in performance than modeling.

I applied pre-processing changes with the `CRCNN` model with default hyperparameters for `semeval` and manual hyperparameters for the medical datasets. All comparisons are performed against the original pre-processing technique, which involved using the original dataset sentences in training and test.

| Preprocess \ Dataset | semeval | ddi | | i2b2 | |
|---|---|---|---|---|---|
| | | Class | Detect | Class | Detect |
| Original | **81.55** | 65.53 | 81.74 | 59.75 | 83.17 |
| | 80.85 (1.31) | 82.23 (0.32) | 88.40 (0.48) | 70.10 (0.85) | 86.45 (0.58) |
| Entity Blinding | 72.73 | **67.02** | **82.37** | **68.76** | **84.37** |
| | 71.31 (1.14) | 83.56 (2.05)• | 89.45 (1.05)• | 76.59 (1.07) | 88.41 (0.37) |
| Punct and Digit | 81.23 | 63.41 | 80.49 | 58.85 | 81.96 |
| | 80.95 (1.21)• | 80.44 (1.77) | 87.52 (0.98) | 69.37 (1.43)• | 85.82 (0.43) |
| Punct, Digit and Stop | 72.92 | 55.87 | 76.57 | 56.19 | 80.47 |
| | 71.61 (1.25) | 78.52 (1.99) | 85.65 (1.21) | 68.14 (2.05)• | 84.84 (0.77) |
| NER Blinding | 81.63 | 57.22 | 79.03 | 50.41 | 81.61 |
| | 80.85 (1.07)• | 78.06 (1.45) | 86.79 (0.65) | 66.26 (2.44) | 86.72 (0.57)• |

Table 4.1: Preprocessing techniques with `CRCNN` model. Row labels Original = simple tokenization and lower casing of words, Punct = punctuation removal, Digit = digit removal and Stop = stop word removal. Test set results at the top with cross validated results (average with standard deviation) below. All cross validated results are statistically significant compared to Original preprocessing ($p < 0.05$) using a paired t-test except those marked with a •

**Punctuation and Digits are important in the biomedical domain**

Removal of punctuation and digits (`punct`) hurts *classification* and *detection* performance for the `ddi` dataset, which is a biomedical dataset. On the other hand, performance on `i2b2` is worse only for the detection task. Statistical significance is not found for the other tasks and datasets.

This indicates that punctuation and digits are more important for the `ddi` dataset and that they are important only for the detection of relations for `i2b2`. To further investigate which of punctuation and digit normalization was the larger contributor in worse performance, I looked at examples where misclassifications were occuring. Details of this analysis are located in appendix B, section B.1.

**Stop words are important in relation extraction settings**

Removal of punctuation, digits and stop words (`stop`) is hurting performance more than `punct` (statistically significant for `ddi` and `semeval` with $p < 0.005$). This effect is less drastic for `i2b2`: `stop` is not statistically significantly worse than `punct` for *classification* task, but is significantly worse with $p = 0.015$ for the *detection* task.

This indicates that stop words are important for relation extraction.

Looking at examples misclassified by `stop` revealed important stop words for different relations in the datasets. This analysis is present in section B.4 of appendix B.

**Fine-grained concept types could be helpful in general because of their ability to simplify the sentence**

The availability of fine-grained concept types is likely to boost performance in relation extraction settings. The `i2b2` dataset provided fine-grained concept types in the form of medical problem, test and treatments. Entity blinding causes almost 9% improvement in *classification* performance and 1% improvement in *detection* performance. In contrast, `ddi` only provided gold standard annotations for drug types in the sentence, and while this does not cause statistically significant improvements for cross validation, it does improve test set classification performance by about 1.5% and detection performance by 1%. For these medical datasets, NER blinding consisted of replacing the detected named entities by *Entity* because named entity types were not available (more details in section 3.1.1 of chapter 3). Due to the coarse-grained nature of the entities, it hurts *classification* performance significantly, and *detection* performance a little. Further investigations into these are located in appendix B in sections B.2 and B.3.

Entity blinding hurts performance for `semeval`, possibly due to the coarse grain nature of the replacement and the entity bias [87]. The replacement loses associations between the entity mentions and relation types, which reduces performance. While a finer-grain replacement in this setting (NER blinding) does not cause a statistically significant change in performance, it has been shown to be a helpful feature by [69]. To recall, entity blinding involved replacement of entity words by *Entity*, while NER blinding involved replacing named entities in the sentence with labels such as *ORGANIZATION* and *PERSON* (more details in section 3.1.1 of chapter 3).

**Reasonable performance is maintained on the *Detection* task**

For the medical datasets, while *classification* performance varies highly with different pre-processing techniques, *detection* is relatively unaffected. In a setting where one cares more about detection of relationships rather than multi-class classification, one would be able to get away with using non-complicated pre-processing techniques to maintain reasonable performance.

### 4.2.2 Split Bias: Why reporting on one test set score is problematic

All 3 datasets evaluate models based on one score on the test set, which is common practice for NLP challenges. Reporting one score as opposed to a distribution of scores has been shown to be problematic by Reimers and Gurevych [59] for sequence tagging. Recently, Crane [14] discuss similar problems for question-answering. My experiments show that even if you keep the same random initialization seed (all our experiments have a fixed random initialization seed), split bias can be another source of variation in scores.

Significance testing of some cross validated results reveals no significance even when the test set result improves in performance. This is particularly concerning for `ddi` where entity blinding (called drug blinding in the literature) is used as a standard pre-processing technique without ablation studies demonstrating its effectiveness. Results suggest the contrary: entity blinding seems to help test set performance for `ddi` in table 4.1, but shows no statistical significance. Table 4.5 further demonstrates that using this in conjunction with other techniques results in test score variations despite being statistically insignificant.

No statistical significance is seen even when the test set result worsens in performance for BERT-CLS in table 4.2 where it hurts test set performance on `ddi` but is not statistically significant when cross validation is performed.

| Modeling \ Dataset | semeval | ddi | | i2b2 | |
|---|---|---|---|---|---|
| | | Class | Detect | Class | Detect |
| CRCNN | 81.55<br>80.85 (1.31) | 65.53<br>82.23 (0.32) | 81.74<br>88.40 (0.48) | 59.75<br>70.10 (0.85) | 83.17<br>86.45 (0.58) |
| Piecewise pool | 81.59<br>80.55 (0.99)• | 63.01<br>81.99 (0.38)• | 80.62<br>88.47 (0.48)• | 60.85<br>73.79 (0.97) | 83.69<br>89.29 (0.61) |
| BERT-tokens | 85.67<br>85.63 (0.83) | **71.97**<br>85.35 (0.53) | **86.53**<br>90.70 (0.46) | 63.11<br>72.06 (1.36) | **84.91**<br>87.57 (0.75) |
| BERT-CLS | 82.42<br>80.83 (1.18)• | 61.3<br>82.71 (0.68)• | 79.63<br>88.35 (0.77)• | 56.79<br>67.37 (1.08) | 81.91<br>85.43 (0.36) |
| ELMo | **85.89**<br>84.79 (1.08) | 66.63<br>84.53 (0.96) | 83.05<br>90.11 (0.56) | **63.18**<br>72.53 (0.80) | 84.54<br>87.81 (0.34) |

Table 4.2: Modeling techniques with original preprocessing. Test set results at the top with cross validated results (average with standard deviation) below. All cross validated results are statistically significant compared to `CRCNN` model ($p < 0.05$) using a paired t-test except those marked with a •. In terms of statistical significance, comparing contextualized embeddings with each other reveals that BERT-tokens is equivalent to ELMo for `i2b2`, but for `semeval` BERT-tokens is better than ELMo and for `ddi` BERT-tokens is better than ELMo only for detection.

### 4.2.3 Modeling

In table 4.2, I tested the generalizability of the commonly used piecewise pooling technique proposed in [82], a variant of which was applied in the model by Luo et al. for `i2b2`. I also tested the improvements offered by different featurizations of contextualized embeddings, which has not been explored much for relation extraction.

Modeling changes were applied with the original pre-processing technique for the `CRCNN` model with default hyperparameters for `semeval` and manual hyperparameters for the medical datasets. All comparisons are performed with the baseline performance of the `CRCNN` model.

#### Piecewise pooling is not a generalizable technique

While piecewise pooling helps `i2b2` by 1%, it hurts test set performance on `ddi` and doesn't affect performance on `semeval`. It may be intuitive to split pooling by entity location, but this technique is not experimentally found to be generalizable to other datasets.

| Hyperparam Tuning \ Dataset | semeval | ddi | | i2b2 | |
|---|---|---|---|---|---|
| | | Class | Detect | Class | Detect |
| Default | 81.55 | 62.55 | 80.29 | 55.15 | 81.98 |
| | 80.85 (1.31) | 81.62 (1.35) | 87.76 (1.03) | 67.28 (1.83) | 86.57 (0.58) |
| Manual Search | - | **65.53** | **81.74** | **59.75** | **83.17** |
| | | 82.23 (0.32)• | 88.40 (0.48)• | 70.10 (0.85) | 86.45 (0.58)• |
| Random Search | **82.2** | 62.29 | 79.04 | 55.0 | 80.77 |
| | 81.10 (1.26)• | 75.43 (1.48) | 83.54 (0.60) | 60.66 (1.43) | 82.73 (0.49) |

Table 4.3: Hyperparameter tuning methods with original preprocessing and fixed `CRCNN` model. Test set results at the top with cross validated results (average with standard deviation) below. All cross validated results are statistically significant compared to Default with $p < 0.05$ except those marked with a •. Note that hyperparameter tuning can involve much higher performance variation depending on the distribution of the data. Therefore, even though there is no statistical significance in the manual search case for the held out fold in the ddi dataset, there was statistical significance for the dev fold which drove those set of hyperparameters. For both ddi and i2b2 datasets, manual search is better than random search with $p < 0.05$.

**Contextualized embeddings should be featurized correctly in CNN models**

Contextualized embeddings generally boost performance, but they should be concatenated with the word embeddings before the convolution stage. ELMo and BERT-tokens boosted performance significantly for all datasets, but BERT-CLS hurt performance for the medical datasets. While BERT-CLS boosted test set performance for `semeval`, this was not found to be a statistically significant difference for cross validation. Note that ELMo was featurized similarly to BERT-tokens and featurization details are present in section 3.1.2 of chapter 3.

This indicates that the technique of featurizing the contextualized embeddings matters for a CNN architecture. Concatenating the contextualized embeddings with the word embeddings keeps a tighter coupling, which is helpful for relation extraction where the word level associations are essential in predicting the relation type.

### 4.2.4 Hyperparameter Tuning

Bergstra and Bengio [5] show the superiority of random search over grid search in terms of faster convergence, but leave to future work automating the procedure of manual tuning, i.e. sequential optimization. Bayesian optimization strategies could

help with this [68] but often require expert knowledge for correct application. I tested how manual tuning, requiring less expert knowledge than Bayesian optimization, would compare to the random search strategy in table 4.3.

#### Manual search outperformed random search

Tables in appendix C demonstrate that random search reduces the variability of results and converges to better performance than the default hyperparameters. Additionally, manual search outperformed random search for both `i2b2` and `ddi` corpus. Both methods present different challenges for barrier of entry.

Manual search is often criticized for the high barrier of entry [5]. Knowledge about which hyperparameters are more important in specific contexts can make this search faster and provide improved results. My proposed two-pass method helps in developing intuition on the important hyperparameters by changing each hyperparameter in isolation to test the statistical significance of the performance difference. By further changing the narrow list of hyperparameters found from the first pass, convergence to better results is found in the second pass.

Random search, on the other hand, can be complicated because one needs to pick the right distributions for the hyperparameters and the right search space. A larger search space and sub-optimal distributions run into the possibility of running too many experiments in a hyperparameter space leading to lower performance. Ideally, random search should run enough experiments in the vicinity of the global maxima to converge to it faster. Additional findings related to result distributions for random search is present in appendix C.

### 4.2.5 Evaluation Metrics

Picking the right evaluation metric is critical, and it is important to choose a metric that has the biggest delta between the performance of different models. Test set results for different evaluation metrics on the pre-processing and modeling techniques are presented in the tables of appendix D.

**The choice of macro or micro statistics is dictated by class imbalance**

When using micro and macro statistics (precision, recall and F1), class imbalance of the dataset dictates the one to pick. Macro statistics are highly affected by imbalance, whereas micro statistics are able to recover well. Despite suffering due to class imbalance, though, macro statistics may be more appropriate than micro as they provide stronger discriminative capabilities by providing equal importance to classes of smaller sizes. However, micro statistics are as discriminative as macro statistics in settings when the classes are relatively balanced. In the next two paragraphs, I will discuss claims pertaining to the *classification* task.

Compared to `semeval`, `ddi` and `i2b2` suffer from stark class imbalances as seen in chapter 2.4. `semeval` has a number of examples in classes (for those affecting the metrics) ranging from 200 or 300 to 1000. Its *Other* class has about 3000 examples which are not included in the official metric calculations. `ddi` has one class with 228 examples, while the others have about 1000 examples. The *None* class has 21,948 examples which is included for the official score calculations. `i2b2` has 5 classes in the 100-500 range, while the others contain about 2000 examples. *None* is the largest class with 19,934 examples which are not included in the official micro F1 score calculation.

Using micro statistics is reasonable for `i2b2` because the highly imbalanced class is not included in the calculations. Therefore, this metric is able to be as discriminative as macro statistics. For example, test set micro F1 between baseline and entity blinding techniques is 59.75 and 68.76, while that for macro F1 is 36.44 and 43.76. In contrast, using micro statistics is a bad idea for `ddi` because the performance on the *None* class would drive most of the predictive results of the model. For example, micro-F1 between baseline and NER blinding is 88.69 and 86.18, whereas macro-F1 is 65.53 and 57.22. `semeval` does not have a stark contrast between micro and macro scores due to *Other* class not being included in the calculation. Using either metric to evaluate models is reasonable for this dataset.

### Precision vs Recall comparison

For both `ddi` and `i2b2` datasets, the delta between macro-Precision (macro-P) and macro-Recall (macro-R) is higher than that between micro-P and micro-R (with precision being higher than recall). For example, the delta is 3.2 for micro statistics comparison, whereas the delta is 15.04 for macro statistics for the baseline result of the `i2b2` dataset. This indicates that recall is not as good compared to precision for relations with fewer examples in the dataset. This is an important consideration for medical settings where the availability of a class-balanced dataset may be difficult. Therefore, those settings looking for deployable models that may value recall higher than precision should evaluate models based on metrics such as Precision-Recall curves in addition to the standard F1 scores calculated for NLP challenge tasks.

### Task comparison

The detection task does not suffer from such variations due to the lower class imbalance. For example, `ddi` dataset micro-F1 between baseline and NER blinding model is 90.01 and 88.74, while macro-F1 is 81.74 and 79.03. This further suggests that modeling differences and pre-processing differences cause more variation in performance in settings when the class imbalance is higher.

### Accuracy and Micro statistics

Finally, accuracy and micro statistics result in the same number when all relation classes trained on are evaluated on. Refer to the accuracy equation in section 3.2.2 of chapter 3, which is equivalent to micro-P by definition. The reason micro-R is equivalent to accuracy is because the sum of False Negatives in a multi-class setting become equivalent to the sum of False Positives for each class (to avoid duplicates). Therefore, in datasets such as `ddi`, evaluating on either should be ok. The other datasets would have similar trends if they had not ignored *Other* and *None* classes from the official evaluation metric calculations.

## 4.3 Additional Experiments

In order to compare my results with current state-of-the-art results for each dataset, I ran additional experiments to test combinations of techniques that showed improvements in section 4.2. These are listed in tables 4.4 and 4.5.

Section 4.2.1 showed that even though the entity blinding technique was not statistically significantly better for `ddi`, it was improving test set performance. The table 4.5 shows the variation in test set performance, and shows that performing entity blinding does not cause statistically significant improvements even when used in combination with other modeling techniques. This further demonstrates that entity blinding is not a helpful pre-processing technique for `ddi`.

In table 4.5, using entity blinding with contextualized embedding helps test set performance for the E + ent row, but hurts test set performance for the B + ent row when compared with the respective contextualized embedding results from table 4.2. However, these results are not statistically significant. This further strengthens the claim made in section 4.2.2 about statistical significance being necessary to gauge the generalizability of a technique across different splits of the same dataset.

#### Comparison to current state of the art methods

The best *classification* test set results found are listed in table 4.6. Note that I do not compare the *extraction* task for datasets other than `ddi` because the official challenges only compared classification results. Even though the official challenge did not rank models based on the *detection* task, recent papers in the `ddi` literature mention these results.

I report results in table 4.6 to perform a comparison to state-of-the-art approaches consistent with the current method, and show why this leads to unfair comparisons. This is not only because of the problem of split bias highlighted in section 4.2.2, but also because different models are using different pre-processing techniques, which are critical sources of variation in results. The issue is more pronounced for the medical datasets, where omission of ablation studies is common as seen in section 1.3.1 of

| Technique \ Task | Classification | Detection |
|---|---|---|
| E + ent | 70.46<br>77.70(1.26) | 86.17<br>89.36 (0.50) |
| B + ent | 70.56<br>76.72 (1.04) | 85.66<br>88.63 (0.33) |
| E + piece + ent | 70.62<br>79.41 (0.53) | 86.14<br>90.37 (0.44) |
| B + piece + ent | **71.01**<br>79.51 (1.09) | **86.26**<br>90.34 (0.53) |
| piece + ent | 69.73<br>78.12 (1.10) | 85.44<br>89.74 (0.44) |
| E + piece | 63.19<br>74.76 (0.68) | 84.92<br>89.90 (0.37) |
| B + piece | 63.23<br>74.67 (0.89) | 85.45<br>89.61 (0.68) |

Table 4.4: Additional experiments for `i2b2`. E = ELMo, B = BERT-tokens, ent = entity blinding, piece = piecewise pooling. All results are statistically significant compared to BERT-tokens and ELMo models respectively from table 4.2 and piece + ent row is statistically significant compared to piecewise pool model as well as entity blinding model. These are all statistically significantly better than the `CRCNN` model from table 4.2. All $p < 0.05$.

chapter 1.

Wang et al. [73] report a result of 88% on `semeval` and do not provide any public source code for replication purposes. Despite being below the state of the art range, `REflex` provides the best performing publicly available model for this dataset.

Zheng et al. [90] report the best result on `ddi` (77.3%) but perform negative instance filtering, which is a highly specific pre-processing technique that does not fit with the flexible nature of `REflex`. This technique also makes the data smaller, but the paper is unclear about whether they apply this technique to shorten the test set as well. Unfortunately, the source code is not publicly available to answer these questions. Additionally, cutting out sentences from the training as well as test data would make the prediction task a lot easier and impractical to use in real-world settings due to its highly specific nature.

Zhao et al. [89] already show that negative instance filtering causes a 4.1% im-

| Technique \ Task | Classification | Detection |
|---|---|---|
| E + ent | 68.69<br>86.25 (1.54) | 83.72<br>91.35 (0.90) |
| B + ent | **70.66**<br>85.79 (1.54) | **85.35**<br>91.26 (0.63) |

Table 4.5: Additional experiments for `ddi`. E = ELMo, B = BERT-tokens, ent = entity blinding. Results are not statistically significant compared to BERT-tokens and ELMo models respectively from table 4.2 and not from each other either.

| Dataset | Result | Technique |
|---|---|---|
| `semeval` | 85.89 | ELMo |
| `ddi` | 71.97, 86.53 | BERT-tokens |
| `i2b2` | 71.01 | BERT-tokens + piece + ent |

Table 4.6: Best test set *classification* results for all datasets, except `ddi` where *detection* results are mentioned after the classification results. piece = Piecewise pooling, ent = entity blinding. Result corresponds to F1 scores, macro for `semeval` and `ddi`, but micro for `i2b2`.

provement in test set performance. If my model were to use this pre-processing technique, it would reach the state-of-the-art range in the *classification* task. On the other hand, my results from the *detection* results **outperform** this model by 2.53%.

Sahu et al. [64] (code unavailable) report a state of the art result of 71.16% on `i2b2`, which the results in table 4.6 are able to match. Note that [61] report a result of 73.7% with a support vector machine, but they used a larger version of the dataset. After the official challenge, only a subset of the data was publicly available, so comparing against this number would not be fair.

Comparison against these numbers demonstrates that `REflex` is the only open-source framework, capable of achieving performance in the state of the art ranges for all 3 datasets I evaluate on. Therefore, `REflex` can be used as a strong baseline model in future relation extraction studies.

# Chapter 5

# Conclusion

Relation Extraction (RE) suffers from an issue with reproducibility and a lack of consensus on generalizable techniques, which make it difficult to perform systematic comparison of methods.

`REflex` is an open-source and extendable framework that will help the community in performing systematic model and model-complementing explorations on new datasets. For the 2 of the 3 datasets `REflex` is applied to, it is the only open source model in the state-of-the-art ranges (shown in section 4.3 of chapter 4). Therefore, the model can also be used as a strong baseline in future RE studies.

My exploration on the 3 datasets reveal variations offered by pre-processing and training methodologies, which often go unreported. This indicates that comparing models without having these techniques standardized can make it difficult to assess the true source of performance gains. The key findings are:

- Pre-processing can have a strong effect on performance, sometimes more than modeling techniques, as is the case of `i2b2`. Concept types seem to offer useful information, perhaps revealing more general semantic information in the sentence that can help with predictions. Fine-grained Gold standard annotated concept types are most beneficial, but those from automatically extracted packages may also be useful as long as they consist of multiple types. Punctuation and digits may hold more importance in biomedical settings, but stop words

hold significance in all settings.

- Reporting on one test set score can be problematic due to split bias, and a cross validation approach with significance tests may help ease some of this bias. Drug blinding for `ddi` is commonly used in the literature but does not seem to offer any statistically significant improvements. Therefore, this technique is unnecessary for this domain.

- Contextualized embeddings are generally helpful but the featurizing technique is important: for CNN models, concatenating contextualized embeddings with the word embeddings before convolution is most beneficial.

- Picking the right hyperparameters for a dataset is important to performance. I suggest an initial manual hyperparameter search based on cross validation significance tests because that may be sufficient in most cases. If one is not pressed for time, random search is a reasonable automated option for hyperparameter tuning, but requires more experience for picking the right search space and the right distributions for the hyperparameters.

- Picking the right evaluation metrics for a new dataset should be driven by class imbalance issues for the classes chosen to be evaluated on.

The problems hindering progress in the RE community would be further eased with appropriate future work highlighted in the following section.

## 5.1 Future Work

`REflex` provides the ability to easily include additional components that could help with future analysis. Further extension to my methods could reveal more insights about the datasets:

- Pre-processing methods can be dissected further to separately test the performance variation with punctuation and digit normalization. Another useful

future extension would be to test whether improvements are offered by NER and gold standard entity type information as additional features.

- More modeling techniques such as long short-term memory networks (LSTM) and plain CNN with cross entropy loss could be tested to compare the ability of both models to capture entity interactions in the sentence. A study similar to [25] could be performed in this case.

- More hyperparameter tuning methods such as Hyperopt [6], Spearmint [68] and Grid Search [5] could be tested.

- More evaluation metrics such as AUPRC (area under the precision-recall curve) could be compared for all modeling and pre-processing techniques.

# Appendix A

# Quantitative Literature Review

In the table on the next page, the following columns are present:

**cite** = number of papers that cited the paper

**code** = whether code was publicly available (y for yes and • for no)

**ablation** = whether an ablation study was performed

**hyperparam** = whether hyperparameter details were mentioned

**cross val** = whether cross validation details were mentioned

**word-embed** = whether information about word embeddings used was mentioned

**datasets** = number of datasets evaluated on

| paper | cite | code | ablation | hyperparam | cross val | word-embed | datasets |
|---|---|---|---|---|---|---|---|
| Socher et al. [69] | 890 | y | • | y | • | y | 2 |
| Zeng et al. [81] | 477 | • | y | y | y | y | 1 |
| Santos et al. [66] | 220 | • | y | y | y | y | 1 |
| Nguyen and Verspoor [51] | 146 | • | y | y | y | • | 2 |
| Miwa and Bansal [50] | 175 | • | y | y | y | • | 3 |
| Li and Jurafsky [36] | 107 | y | y | y | • | y | 6 |
| Xu et al. [76] | 108 | • | y | y | • | y | 1 |
| Wang et al. [73] | 102 | • | y | • | • | y | 1 |
| Hashimoto et al. [19] | 64 | • | y | y | • | y | 1 |
| Zhang and Wang [83] | 68 | • | y | • | y | y | 2 |
| Vu et al. [72] | 57 | • | y | y | • | y | 1 |
| Yin et al. [79] | 116 | • | n | y | • | • | 7 |
| Yu et al. [80] | 45 | y | y | y | y | y | 1 |
| Xu et al. [78] | 54 | y | y | y | • | • | 1 |
| Zhang et al. [84] | 51 | • | • | • | • | y | 1 |
| Nguyen and Grishman [52] | 42 | • | y | y | • | y | 2 |
| Qin et al. [55] | 39 | • | • | y | y | y | 1 |
| Cai et al. [8] | 44 | • | y | y | • | y | 1 |
| Sahu et al. [64] | 32 | • | y | y | y | y | 1 |

| Paper | cite | code | ablation | hyperparam | cross val | word-embed | datasets |
|---|---|---|---|---|---|---|---|
| Adel et al. [1] | 29 | y | y | • | • | y | 1 |
| Zeng et al. [82] | 190 | • | y | y | • | y | 1 |
| Xu et al. [77] | 171 | • | y | y | • | y | 1 |
| Zhang et al. [88] | 3 | • | y | y | • | y | 2 |
| Levy et al. [34] | 20 | y | y | y | • | y | 1 |
| Liu et al. [43] | 48 | • | • | y | • | y | 1 |
| Zhao et al. [89] | 41 | y | y | y | • | y | 1 |
| Ebrahimi and Dou [15] | 30 | • | • | • | • | • | 2 |
| Li et al. [35] | 27 | y | y | y | y | y | 2 |
| Quan et al. [56] | 23 | y | • | y | y | y | 2 |
| Sahu and Anand [65] | 13 | y | y | y | • | y | 1 |
| Liu et al. [42] | 9 | • | • | y | • | y | 1 |
| Lim and Kang [40] | 4 | • | • | • | • | • | 1 |
| Zheng et al. [90] | 12 | • | y | y | y | y | 1 |
| Wang et al. [75] | 5 | n | y | y | • | y | 1 |
| Lim et al. [41] | 1 | y | y | y | y | y | 2 |
| Kavuluru et al. [30] | 8 | • | • | y | • | • | 1 |
| Huang et al. [23] | 4 | • | • | y | • | y | 1 |
| Juan Hou and Ceesay [27] | 1 | • | • | • | • | y | 1 |

| **Paper** | **cite** | **code** | **ablation** | **hyperparam** | **cross val** | **word-embed** | **datasets** |
|---|---|---|---|---|---|---|---|
| Lim and Kang [39] | 4 | y | • | y | • | y | 1 |
| Rotsztejn et al. [62] | 2 | • | • | y | y | y | 1 |
| Jin et al. [26] | 0 | • | y | y | y | y | 1 |
| Sahu et al. [64] | 31 | • | y | y | y | y | 1 |
| Luo [46] | 21 | • | • | y | • | y | 1 |
| Lv et al. [48] | 15 | • | • | • | • | • | 1 |
| Jin et al. [26] | 14 | • | y | y | • | y | 1 |
| Chikka and Karlapalem [11] | 1 | y | • | y | • | • | 1 |
| Li et al. [38] | 0 | y | • | y | y | y | 1 |
| Li et al. [37] | 0 | • | • | • | • | • | 5 |
| Suster et al. [70] | 0 | y | • | y | • | y | 1 |
| Luo et al. [47] | 16 | y | • | y | • | y | 1 |
| He et al. [21] | 2 | • | • | y | • | y | 1 |
| He et al. [20] | 0 | • | • | y | y | y | 2 |
| Nguyen and Verspoor [51] | 1 | • | y | y | • | y | 1 |

Table A.1: Quantitative Literature Review

# Appendix B

# Further investigation into misclassified examples for Pre-processing techniques

Section 4.2.1 in chapter 4 contains a prelude to this analysis.

## B.1 Which of Punctuation and Digit Removal are important for the medical datasets?

I gathered statistical information about the examples where punctuation and digit removal (`punct`) led to an incorrect prediction, but original pre-processing (`original`) led to a correct prediction, in table B.1. From these examples, numbers were blinded in only 31 out of 150 sentences for `ddi`. This indicates that removal of punctuation is driving the worse performance from this pre-processing technique for `ddi`.

For `i2b2`, only *detection* task was being hurt by `punct`, which is also seen in table B.1 where *None* is contributing to most misclassifications. 399 sentences out of 963 had the numbers blinded here, which does not help in decoupling the effects of punctuation versus digit normalization in the performance.

| Dataset | Total | Most misclass relation | Digit normalization | |
|---|---|---|---|---|
| | | | Mean ± std | Other stats |
| ddi | 150 | *None* (85) | 0.56 ± 1.7 | (16,0,0) |
| i2b2 | 963 | *None* (649) | 4.78 ± 2.97 | (31,2,4) |

Table B.1: Statistics on misclassified examples. Total = total misclassified examples, Most misclass relation = relation that is most incorrectly predicted with number of examples, Mean ± std = average and standard dev of number of digits that are normalized per sentence, Other stats = (Max, Min, Median) of the number of digits normalized per sentence. The total column represents about 3% of the test data for ddi and 5% for i2b2.

## B.2 Why does entity blinding help i2b2?

A further investigation into why entity blinding was improving classification performance on i2b2 was necessary to prove or disprove my hypothesis that entity blinding is helpful because of its ability to simplify the sentence.

First, I investigated whether quantitative information about entity overlap and sentence and context length could have an effect on this result. For example, if i2b2 had an unusually low train and test overlap compared to the other datasets, the entity mentions would not have an effect in relation prediction. This would mean that entity blinding was not hurting performance due to the specific way this dataset was constructed and would weaken the hypothesis of fine-grained entity blinding being helpful in general. Similarly, if sentence and context length was unusually low for i2b2 compared to the other datasets, the entity mentions themselves would be larger contributors to relation prediction.

Second, I performed a qualitative analysis by looking at the examples for which the model with entity blinding (ent) was making correct relation predictions, but the one with the original pre-processing technique (original) was making incorrect relation predictions.

- The first quantitative finding was the percentage of test examples that had overlapping entities (those that had a relation label including *None* and *Other*) with the training data, and the proportion of the training data that these examples constituted. This is listed in table B.2.

| Dataset | Test overlap | Train overlap |
|---|---|---|
| semeval | 8.5 | 3.5 |
| ddi | 28.59 | 13.78 |
| i2b2 | 7.73 | 11.55 |

Table B.2: Percentage of overlapping entities. Test overlap is the percentage of test examples with overlapping entities from the train data, whereas train overlap is the percentage of training examples the overlapping entities were present in.

The second quantitative finding was related to the average length of context and sentence in all datasets. This information is listed in table B.3

| Dataset | Sentence length | Context length |
|---|---|---|
| semeval | 20 | 5 |
| ddi | 33 | 11 |
| i2b2 | 48 | 15 |

Table B.3: Average sentence and context length of the datasets. Context length refers to the number of words between the entities, including the entity words themselves.

Quantitative comparison of the datasets revealed that at least one other dataset had similar statistics from comparisons in tables B.2 and B.3. Because entity blinding was only helpful for i2b2 even though other datasets had similar quantitative information, I concluded that these findings did not weaken my original hypothesis.

- The qualitative study revealed a total of 1815 examples (9.5% of the test data) for which ent was making correct predictions but original was making incorrect predictions. All sentences involved entities that were blinded and *None* relation was incorrectly predicted the most (706 examples), followed by *PIP* and *TeRP*. Overall, ent prevents more *False Negatives*: it allows correct prediction of the relation label whereas original predicts the same examples as *None*.

  i2b2 also divides relations into 3 disjoint higher level categories: *Problem-Problem*, *Problem-Test* and *Test-Treatment*. Knowledge of entity types should narrow down possible relation types to lead to fewer chances of error.

  2 common types of examples were found for incorrect predictions by the baseline

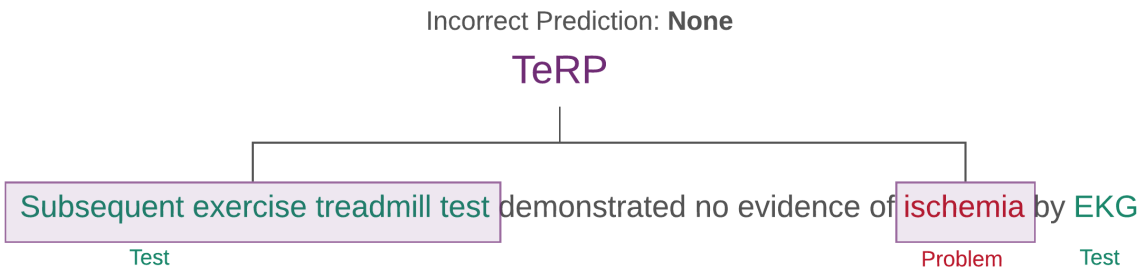


Figure B-1: Correct prediction being *TeRP* i.e. Test reveals Medical Problem, and baseline model predicts *None* incorrectly. Periods omitted for presentation. Those entities marked with Test and Problem are blinded by the entity blinding pre-processing technique.

model:

1. `original` predicting *None* relation. In such cases, shortening of the sentence by blinding might allow the model to focus highly on an indicative context word. An example is present in figure B-1. In many such examples, the word *demonstrate* appears multiple times in the context, and it is likely to be highly indicative of the *TeRP* relation. Even for examples where `original` incorrectly predicts a non-*None* relation, shortening of the sentence via blinding seems helpful in focusing on meaningful context words.

2. `original` predicting an impossible relation. For example, *PIP* relation can only exist between two medical problems. However, the baseline model incorrectly predicts *PIP* for relations between medical problems and tests. Figure B-2 demonstrates such an example.

## B.3 Why does NER blinding hurt performance on the medical datasets?

To investigate the examples where NER blinding was leading to incorrect predictions, but original pre-processing led to correct predictions, I gathered statistics about the examples in table B.4. All sentences in the table consisted of entities that were blinded.

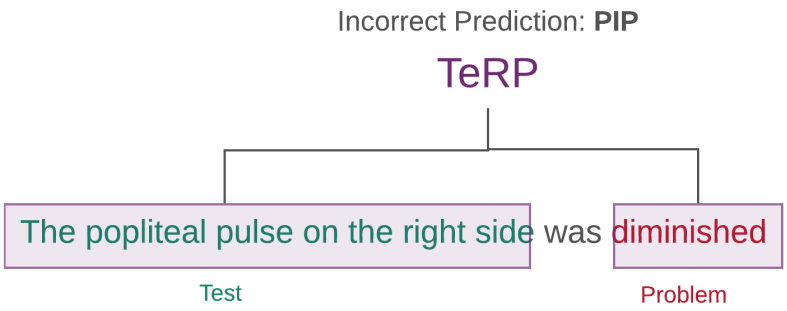


Figure B-2: Correct prediction being Test reveals Medical Problem, and baseline model predicts *PIP* incorrectly. Periods omitted for presentation. Those entities marked with Test and Problem are blinded by the entity blinding pre-processing technique.

As seen in figures B-3 and B-4, the NER blinding technique for medical datasets is too general, and will remove important words that contribute to the correct relation prediction. [1]

| **Dataset** | **Total** | **Most misclass relation** | **Entity blinding** | |
|---|---|---|---|---|
| | | | Mean ± std | Other stats |
| `ddi` | 311 | *None* (155) | 14.24 ± 8.17 | (41,4,12) |
| `i2b2` | 2024 | *None* (818) | 8.39 ± 5.82 | (48,1,7) |

Table B.4: Statistics on misclassified examples. Total = total misclassified examples, Most misclass relation = relation that is most incorrectly predicted with number of examples, Mean ± std = average and standard dev of number of entities that are blinded per sentence, Other stats = (Max, Min, Median) of the number of entities blinded per sentence. The total column represents about 6.6% of the test data for `ddi` and 10.6% for `i2b2`.

## B.4 Which stop words are important to different relations in the datasets?

Looking at the examples where stop word removal (`stop`) led to an incorrect prediction, but original pre-processing (`original`) led to the correct prediction revealed that some stop words are important for specific relation predictions.

[1] Consequently, it is possible that using NER as a feature might be helpful in making the model focus on these words important to the relation prediction

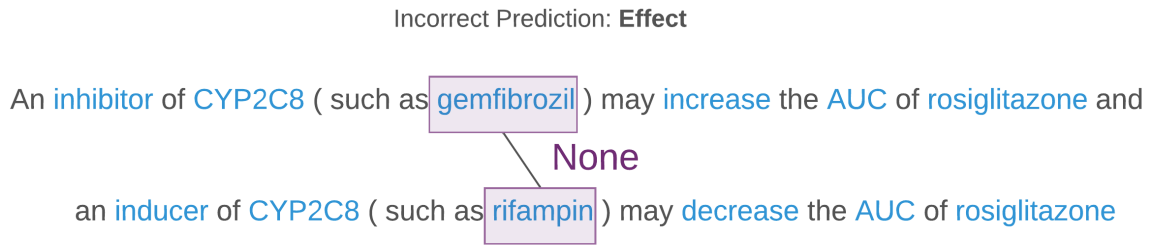


Figure B-3: Correct prediction being *None*, and model using NER blinding predicts *Effect* incorrectly. Periods omitted for presentation. The text colored in blue is blinded to *ENTITY* by the blinding.

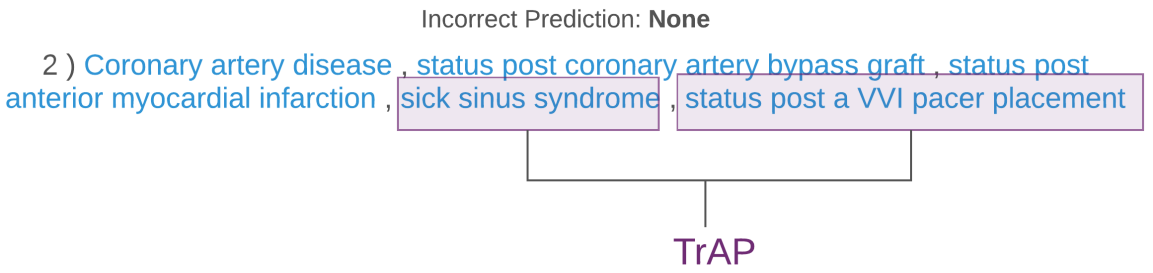


Figure B-4: Correct prediction being *TrAP* i.e. Treatment is administered for Medical Problem, and model using NER blinding predicts *None* incorrectly. Periods omitted for presentation. The text colored in blue is blinded to *ENTITY* by the blinding.

### B.4.1 Important stop words for `semeval`

364 examples (13.4% of the test data) had `stop` being harmful, but `original` leading to correct predictions. The results in table 4.1 showed that punctuations and digits were not harmful, but that `stop` was harmful. This indicates that removal of stop words contributed to the worse performance of `stop` compared to `original`. *Other* was the most incorrectly predicted relation, with 68 examples, followed by *Entity-Origin(e1,e2)* and *Entity-Destination(e1,e2)*.

In figure B-5, the removal of stop words shortens the sentence to focus more on the word *post*, which likely leads to the incorrect prediction of *Message-Topic(e2,e1)*.

*Entity-Origin(e1,e2)* and *Entity-Destination(e1,e2)* are location dependent and the elimination of stop words such as *from*, *into* and *put* seem to be causing misclassifications. Examples are shown in figures B-5, B-6 and B-7. Specifically for example B-7, the removal of the word *put* changes the meaning of the phrase *put inside* from an action oriented location phrase to *inside*, which is a passive word implying

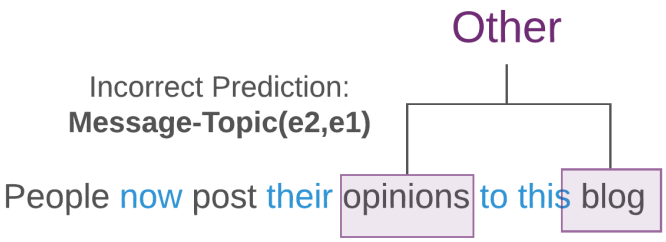


Figure B-5: Correct prediction being *Other*, and model using stop word removal predicts *Message-Topic(e2,e1)* incorrectly. Periods omitted for presentation. The text colored in blue are removed by the stop word removal technique.

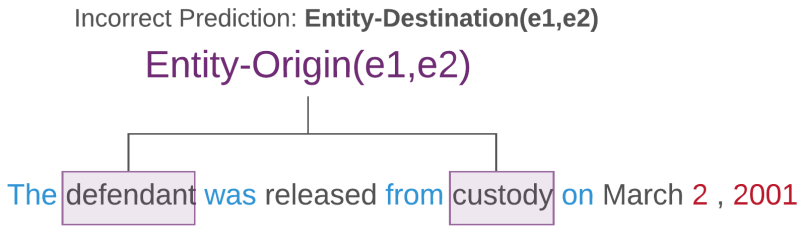


Figure B-6: Correct prediction being *Entity-Origin(e1,e2)*, and model using stop word removal predicts *Entity-Destination(e1,e2)* incorrectly. Periods omitted for presentation. The text colored in blue are removed by the stop word removal technique and those in red are normalized to *NUMBER*.

containment. This could explain why *Entity-Destination(e1,e2)* was misclassified to *Content-Container(e1,e2)*.

### B.4.2 Important stop words for `ddi`

For this corpus, 267 sentences (5.7% of test data) were present where `stop` was harmful, but `original` was helpful. *None* was the most incorrectly predicted relation, with 135 examples, followed by *Mechanism* and *Effect*.

The most common stop word removed from the context was *not*, whose removal likely triggered the prediction changing from *None* to another relation. As seen in figure B-8, the removal of *does not* leaves the context with *affect the clearance...* which likely leads to the incorrect *Mechanism* prediction.

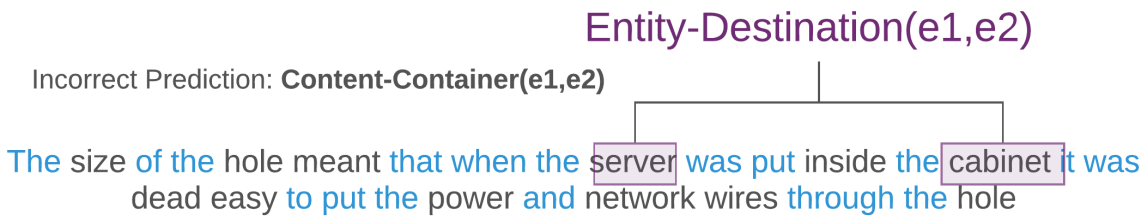


Figure B-7: Correct prediction being *Entity-Destination(e1,e2)*, and model using stop word removal predicts *Content-Container(e1,e2)* incorrectly. Periods omitted for presentation. The text colored in blue are removed by the stop word removal technique.

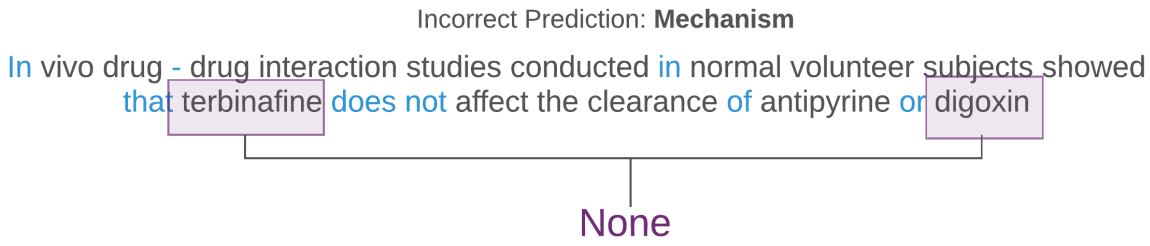


Figure B-8: Correct prediction being *None*, and model using stop word removal predicts *Mechanism* incorrectly. Periods omitted for presentation. The text colored in blue are removed by the stop word removal technique.

### B.4.3 Important stop words for `i2b2`

1399 sentences (7.32% of the test data) had `stop` being harmful, but `original` being helpful. *None* was the most incorrectly predicted relation, with 776 examples, followed by *PIP* and *TrAP*. There seemed to be no common stop words leading to misclassification of the *None* relation, but there non-*None* relations were being misclassified to *None* due to stop words such as *by*, *and* and *with*. One example is shown in figure B-9.

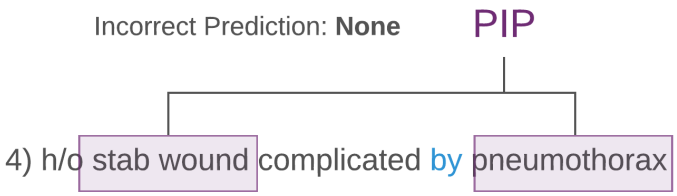


Figure B-9: Correct prediction being *PIP*, and model using stop word removal predicts *None* incorrectly. Periods omitted for presentation. The text colored in blue are removed by the stop word removal technique.

# Appendix C

# Random Search result distributions

For random search, the exact number of experiments run on each dataset differed due to variability in the availability of computation time. A total of `107` experiments were run for `semeval`, `104` for `ddi` and `134` for `i2b2`. Statistics for performance on the randomly sampled dev set are present in tables C.1, C.2 and C.3.

| Statistic \ Search Subset | **All** | **Top 10%** |
|---|---|---|
| Mean | 76.83 | 80.87 |
| Stddev | 9.93 | 0.31 |
| Median | 79.42 | 80.74 |
| Max | 81.37 | 81.37 |
| Min | 4.73 | 80.54 |
| Range | 76.64 | 0.83 |

Table C.1: Random Search experiment statistics for `semeval`. The two columns All and Top 10% determine the subset of the results statistics are gathered for. All = distribution of Macro-F1 scores over 107 experiments, top 10% = distribution over top 10% of the results.

These tables demonstrate that random search reduces the variability of results and converges to better performance than the default hyperparameters.

| Statistic \ Search Subset | **All** | **Top 10%** |
|---|---|---|
| Mean | 80.24 | 82.08 |
| Stddev | 1.63 | 0.25 |
| Median | 80.45 | 82.04 |
| Max | 82.57 | 82.57 |
| Min | 71.21 | 81.74 |
| Range | 11.36 | 0.83 |

Table C.2: Random Search experiment statistics for `ddi`. The two columns All and Top 10% determine the subset of the results statistics are gathered for. All = distribution of Macro-F1 scores over 104 experiments, top 10% = distribution over top 10% of the results.

| Statistic \ Search Subset | **All** | **Top 10%** |
|---|---|---|
| Mean | 69.61 | 72.19 |
| Stddev | 1.54 | 0.39 |
| Median | 69.78 | 72.13 |
| Max | 72.86 | 72.86 |
| Min | 62.92 | 71.64 |
| Range | 9.94 | 1.22 |

Table C.3: Random Search experiment statistics for `i2b2`. The two columns All and Top 10% determine the subset of the results statistics are gathered for. All = distribution of Macro-F1 scores over 134 experiments, top 10% = distribution over top 10% of the results.

# Appendix D

# Evaluation Metric Results on Test Data

Following are the results calculated with different evaluation metrics pertaining to the discussion in section 4.2.5 of chapter 4. Each row represents a pre-processing, modeling technique or combination based on the additional experiments run on each dataset. Only test set results (as opposed to cross validation) are reported for ease of analysis of the large table. In all the tables, Baseline refers to the `CRCNN` model with original pre-processing and default hyperparameters for `semeval` and manual hyperparameters for the medical datasets (`ddi` and `i2b2`). The following short forms are used to refer to:

**B** = BERT-tokens (only used for the medical datasets tables)

**E** = ELMo (only used for the medical datasets tables)

**Ent Blind** = Entity Blinding (only used for `i2b2` table)

**Piece Pool** = Piecewise Pooling (only used for `i2b2` table)

## D.1 `semeval` Dataset

| Technique \ Metric | **acc** | **micro-P** | **micro-R** | **micro-F1** | **macro-P** | **macro-R** | **macro-F1** |
|---|---|---|---|---|---|---|---|
| Baseline | 77.11 | 79.95 | 85.11 | 82.45 | 79.25 | 84.06 | 81.55 |
| Entity Blinding | 67.94 | 70.72 | 77.15 | 73.8 | 69.77 | 76.31 | 72.73 |
| Punct and Digit | 76.48 | 79.19 | 85.42 | 82.19 | 78.33 | 84.51 | 81.23 |
| Punct, Digit and Stop | 68.28 | 73.0 | 74.78 | 73.88 | 72.84 | 73.48 | 72.92 |
| NER Blinding | 77.25 | 79.3 | 86.03 | 82.53 | 78.49 | 85.13 | 81.63 |
| Piecewise pool | 77.0 | 79.54 | 85.55 | 82.44 | 78.86 | 84.71 | 81.59 |
| ELMo | 77.77 | 81.87 | 84.62 | 83.22 | 81.24 | 83.71 | 82.42 |
| BERT-CLS | 77.77 | 81.87 | 84.62 | 83.22 | 81.24 | 83.71 | 82.42 |
| BERT-tokens | 81.3 | 86.63 | 86.74 | 86.69 | 86.08 | 85.61 | 85.67 |

Table D.1: Different Evaluation Metric results on test set of `semeval` dataset. Only test set results are reported for ease of analysis.

## D.2 `ddi` Dataset

| Technique \ Metric | **acc** | | **micro-P** | | **micro-R** | | **micro-F1** | | **macro-P** | | **macro-R** | | **macro-F1** | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Class | Detect | Class | Detect | Class | Detect | Class | Detect | Class | Detect | Class | Detect | Class | Detect |
| Baseline | 88.69 | 90.01 | 88.69 | 90.01 | 88.69 | 90.01 | 88.69 | 90.01 | 72.32 | 82.06 | 63.48 | 81.43 | 65.53 | 81.74 |
| Entity Blinding | 89.22 | 90.44 | 89.22 | 90.44 | 89.22 | 90.44 | 89.22 | 90.44 | 71.26 | 82.99 | 64.63 | 81.79 | 67.02 | 82.37 |
| Punct and Digit | 88.31 | 89.61 | 88.31 | 89.61 | 88.31 | 89.61 | 88.31 | 89.61 | 69.49 | 81.7 | 60.81 | 79.43 | 63.41 | 80.49 |
| Punct, Digit and Stop | 86.58 | 87.86 | 86.58 | 87.86 | 86.58 | 87.86 | 86.58 | 87.86 | 67.4 | 78.59 | 52.72 | 74.98 | 55.87 | 76.57 |
| NER Blinding | 86.18 | 88.74 | 86.18 | 88.74 | 86.18 | 88.74 | 86.18 | 88.74 | 59.13 | 79.9 | 55.93 | 78.24 | 57.22 | 79.03 |
| Piecewise pool | 88.14 | 89.54 | 88.14 | 89.54 | 88.14 | 89.54 | 88.14 | 89.54 | 70.49 | 81.39 | 60.38 | 79.91 | 63.01 | 80.62 |
| E | 89.76 | 90.97 | 89.76 | 90.97 | 89.76 | 90.97 | 89.76 | 90.97 | 73.41 | 84.36 | 63.65 | 81.9 | 66.63 | 83.05 |
| BERT-CLS | 87.84 | 89.05 | 87.84 | 89.05 | 87.84 | 89.05 | 87.84 | 89.05 | 68.2 | 80.51 | 59.31 | 78.84 | 61.3 | 79.63 |
| B | 91.31 | 92.72 | 91.31 | 92.72 | 91.31 | 92.72 | 91.31 | 92.72 | 77.66 | 87.34 | 69.27 | 85.78 | 71.97 | 86.53 |
| E + Entity Blinding | 89.97 | 91.18 | 89.97 | 91.18 | 89.97 | 91.18 | 89.97 | 91.18 | 72.44 | 84.42 | 66.41 | 83.06 | 68.69 | 83.72 |
| B + Entity Blinding | 90.93 | 92.15 | 90.93 | 92.15 | 90.93 | 92.15 | 90.93 | 92.15 | 76.79 | 86.57 | 63.39 | 84.26 | 70.66 | 85.35 |

Table D.2: Different Evaluation Metric results on test set of `ddi` dataset. Only test set results are reported for ease of analysis.

## D.3 i2b2 Dataset

| Technique \ Metric | acc | | micro-P | | micro-R | | micro-F1 | | macro-P | | macro-R | | macro-F1 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Class | Detect | Class | Detect | Class | Detect | Class | Detect | Class | Detect | Class | Detect | Class | Detect |
| Baseline | 78.68 | 83.17 | 61.39 | 83.17 | 58.19 | 83.17 | 59.75 | 83.17 | 49.24 | 81.16 | 34.2 | 80.29 | 36.44 | 80.69 |
| Entity Blinding | 81.92 | 84.37 | 68.88 | 84.37 | 68.65 | 84.37 | 68.76 | 84.37 | 53.33 | 82.32 | 40.72 | 82.27 | 43.76 | 82.29 |
| Punct and Digit | 77.25 | 81.96 | 58.09 | 81.96 | 59.64 | 81.96 | 58.85 | 81.96 | 49.28 | 79.53 | 33.56 | 79.92 | 34.93 | 79.71 |
| Punct, Digit and Stop | 76.05 | 80.47 | 57.15 | 80.47 | 55.27 | 80.47 | 56.19 | 80.47 | 43.26 | 77.96 | 31.16 | 77.47 | 32.99 | 77.7 |
| NER Blinding | 75.12 | 81.61 | 52.58 | 81.61 | 48.42 | 81.61 | 50.41 | 81.61 | 39.44 | 79.45 | 26.3 | 78.17 | 29.15 | 78.73 |
| Piecewise pool | 78.63 | 83.69 | 59.41 | 83.69 | 62.37 | 83.69 | 60.85 | 83.69 | 46.16 | 81.41 | 35.77 | 82.17 | 36.44 | 81.76 |
| E | 80.4 | 84.54 | 64.56 | 84.54 | 61.86 | 84.54 | 63.18 | 84.54 | 59.28 | 82.69 | 36.17 | 81.97 | 38.1 | 82.31 |
| BERT-CLS | 76.94 | 81.91 | 57.66 | 81.91 | 55.95 | 81.91 | 56.79 | 81.91 | 49.88 | 76.61 | 32.4 | 79.15 | 34.05 | 79.37 |
| B | 80.79 | 84.91 | 64.92 | 84.91 | 61.4 | 84.91 | 63.11 | 84.91 | 58.05 | 83.08 | 36.8 | 82.1 | 39.31 | 82.55 |
| E + Entity Blinding | 83.62 | 86.17 | 72.43 | 86.17 | 68.6 | 86.17 | 70.46 | 86.17 | 60.79 | 84.65 | 40.11 | 83.67 | 42.99 | 84.13 |
| E + Piece Pool + Ent Blind | 83.46 | 86.14 | 71.11 | 86.14 | 70.14 | 86.14 | 70.62 | 86.14 | 54.87 | 84.37 | 42.41 | 84.13 | 44.43 | 84.25 |
| Ent Blind + Piece Pool | 82.72 | 85.44 | 69.49 | 85.44 | 69.98 | 85.44 | 69.73 | 85.44 | 48.82 | 83.49 | 41.97 | 83.61 | 42.89 | 83.55 |
| E + Piece Pool | 80.1 | 84.92 | 61.98 | 84.92 | 64.45 | 84.92 | 63.19 | 84.92 | 49.68 | 82.79 | 36.91 | 83.43 | 37.52 | 83.09 |
| B + Ent Blind | 83.27 | 85.66 | 71.52 | 85.66 | 69.63 | 85.66 | 70.56 | 85.66 | 55.62 | 83.9 | 38.82 | 83.44 | 41.83 | 83.66 |
| B + Ent Blind + Piece pool | 83.57 | 86.26 | 70.9 | 86.26 | 71.13 | 86.26 | 71.01 | 86.26 | 55.6 | 84.43 | 42.58 | 84.49 | 44.4 | 84.46 |
| B + Piece pool | 80.59 | 85.45 | 63.08 | 85.45 | 63.39 | 85.45 | 63.23 | 85.45 | 56.01 | 83.51 | 36.84 | 83.59 | 38.84 | 83.55 |

Table D.3: Different Evaluation Metric results on test set of i2b2 dataset. Only test set results are reported for ease of analysis.

# Bibliography


[1] Heike Adel, Benjamin Roth, and Hinrich Schütze. Comparing convolutional neural networks to traditional models for slot filling. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 828–838. Association for Computational Linguistics, 2016.

[2] Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. Semeval 2017 task 10: Scienceie - extracting keyphrases and relations from scientific publications. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 546–555. Association for Computational Linguistics, 2017.

[3] Nguyen Bach and Sameer Badaskar. A survey on relation extraction. "`http://www.cs.cmu.edu/~nbach/papers/A-survey-on-Relation-Extraction-Slides.pdf`", 2007. [Online; accessed 30-Dec-2018].

[4] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

[5] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305, 2012.

[6] James Bergstra, Dan Yamins, and David D Cox. Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms. In *Proceedings of the 12th Python in Science Conference*, pages 13–20. Citeseer, 2013.

[7] Jari Björne and Tapio Salakoski. Tees 2.2: biomedical event extraction for diverse corpora. *BMC bioinformatics*, 16(16):S4, 2015.

[8] Rui Cai, Xiaodong Zhang, and Houfeng Wang. Bidirectional recurrent convolutional neural network for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 756–765, 2016.

[9] Jose Camacho-Collados and Mohammad Taher Pilehvar. On the role of text preprocessing in neural network architectures: An evaluation study on text categorization and sentiment analysis. In *Proceedings of the 2018 EMNLP Workshop*

*BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 40–46, Brussels, Belgium, 2018. Association for Computational Linguistics.

[10] Gavin C Cawley and Nicola LC Talbot. On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, 11(Jul):2079–2107, 2010.

[11] Veera Raghavendra Chikka and Kamalakar Karlapalem. A hybrid deep learning approach for medical relation extraction. *CoRR*, 2018.

[12] Marc Claesen and Bart De Moor. Hyperparameter search in machine learning. *arXiv preprint arXiv:1502.02127*, 2015.

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

[14] Matt Crane. Questionable answers in question answering research: Reproducibility and variability of published results. *Transactions of the Association of Computational Linguistics*, 6:241–252, 2018.

[15] Javid Ebrahimi and Dejing Dou. Chain based rnn for relation classification. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1244–1249, 2015.

[16] Antske Fokkens, Marieke Van Erp, Marten Postma, Ted Pedersen, Piek Vossen, and Nuno Freire. Offspring from reproduction problems: What replication failure teaches us. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1691–1701, 2013.

[17] Kata Gábor, Davide Buscaldi, Anne-Kathrin Schumann, Behrang QasemiZadeh, Haifa Zargayouna, and Thierry Charnois. Semeval-2018 task 7: Semantic relation extraction and classification in scientific papers. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 679–688, 2018.

[18] Roxana Girju, Preslav Nakov, Vivi Nastase, Stan Szpakowicz, Peter Turney, and Deniz Yuret. Semeval-2007 task 04: Classification of semantic relations between nominals. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 13–18. Association for Computational Linguistics, 2007.

[19] Kazuma Hashimoto, Makoto Miwa, Yoshimasa Tsuruoka, and Takashi Chikayama. Simple customization of recursive neural networks for semantic relation classification. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1372–1376, 2013.

[20] Bin He, Yi Guan, and Rui Dai. Convolutional gated recurrent units for medical relation classification. *CoRR*, abs/1807.11082, 2018.

[21] Bin He, Yi Guan, and Rui Dai. Classifying medical relations in clinical text via convolutional neural networks. *Artificial intelligence in medicine*, 2018.

[22] Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, pages 94–99. Association for Computational Linguistics, 2009.

[23] Degen Huang, Zhenchao Jiang, Li Zou, and Lishuang Li. Drug-drug interaction extraction from biomedical literature using support vector machine and long short term memory networks. *Information Sciences*, 415, 06 2017.

[24] Darrel C Ince, Leslie Hatton, and John Graham-Cumming. The case for open computer programs. *Nature*, 482(7386):485, 2012.

[25] Alon Jacovi, Oren Sar Shalom, and Yoav Goldberg. Understanding convolutional neural networks for text classification. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Brussels, Belgium, 2018. Association for Computational Linguistics.

[26] Di Jin, Franck Dernoncourt, Elena Sergeeva, Matthew McDermott, and Geeticka Chauhan. Mit-medg at semeval-2018 task 7: Semantic relation classification via convolution neural network. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 798–804. Association for Computational Linguistics, 2018.

[27] Wen Juan Hou and Bamfa Ceesay. Extraction of drug-drug interaction using neural embedding. *Journal of Bioinformatics and Computational Biology*, 16, 2018.

[28] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665. Association for Computational Linguistics, 2014.

[29] Tian Kang, Shaodian Zhang, Youlan Tang, Gregory W Hruby, Alexander Rusanov, Noémie Elhadad, and Chunhua Weng. Eliie: An open-source information extraction system for clinical trial eligibility criteria. *Journal of the American Medical Informatics Association*, 24(6):1062–1071, 2017.

[30] Ramakanth Kavuluru, Anthony Rios, and Tung Tran. Extracting drug-drug interactions with word and character-level recurrent neural networks. *2017 IEEE International Conference on Healthcare Informatics (ICHI)*, pages 5–12, 2017.

[31] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

[32] Ji Young Lee, Franck Dernoncourt, and Peter Szolovits. Mit at semeval-2017 task 10: Relation extraction with convolutional neural networks. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 978–984. Association for Computational Linguistics, 2017.

[33] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: pre-trained biomedical language representation model for biomedical text mining. *arXiv preprint arXiv:1901.08746*, 2019.

[34] Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342. Association for Computational Linguistics, 2017.

[35] Fei Li, Meishan Zhang, Guohong Fu, and Donghong Ji. A neural joint model for entity and relation extraction from biomedical text. *BMC bioinformatics*, 18(1): 198, 2017.

[36] Jiwei Li and Dan Jurafsky. Do multi-sense embeddings improve natural language understanding? *arXiv preprint arXiv:1506.01070*, 2015.

[37] Q. Li, Z. Yang, L. Luo, L. Wang, Y. Zhang, H. Lin, J. Wang, L. Yang, K. Xu, and Y. Zhang. A multi-task learning based approach to biomedical entity relation extraction. In *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 680–682, 2018.

[38] Yifu Li, Ran Jin, and Yuan Luo. Classifying relations in clinical narratives using segment graph convolutional and recurrent neural networks (seg-gcrns). *Journal of the American Medical Informatics Association*, 26(3):262–268, 2018.

[39] Sangrak Lim and Jaewoo Kang. ChemicalâĂŞgene relation extraction using recursive neural network. In *Database*, 2018.

[40] Sangrak Lim and Jaewoo Kang. Drug drug interaction extraction from the literature using a recursive neural network. In *PloS one*, 2018.

[41] Sangrak Lim, Kyubum Lee, and Jaewoo Kang. Drug drug interaction extraction from the literature using a recursive neural network. *Plos one*, 13:1–17, 2018.

[42] Shengyu Liu, Kai Chen, Qingcai Chen, and Buzhou Tang. Dependency-based convolutional neural network for drug-drug interaction extraction. *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1074–1080, 2016.

[43] Shengyu Liu, Buzhou Tang, Qingcai Chen, and Xiaolong Wang. Drug-drug interaction extraction via convolutional neural networks. *Computational and mathematical methods in medicine*, 2016, 2016.

[44] Yudong Liu, Zhongmin Shi, and Anoop Sarkar. Exploiting rich syntactic information for relation extraction from biomedical articles. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 97–100. Association for Computational Linguistics, 2007.

[45] HP Luhn. 11 keyword-in-context index for technical literature (kwic index). *Readings in automatic language processing*, 1:159, 1966.

[46] Yuan Luo. Recurrent neural networks for classifying relations in clinical notes. *Journal of Biomedical Informatics*, 72, 07 2017.

[47] Yuan Luo, Yu Cheng, Özlem Uzuner, Peter Szolovits, and Justin Starren. Segment convolutional neural networks (seg-cnns) for classifying relations in clinical notes. *Journal of the American Medical Informatics Association*, 25(1):93–98, 2017.

[48] Xinbo Lv, Yi Guan, Jinfeng Yang, and Jiawei Wu. Clinical relation extraction with deep learning. In *International Journal of Hybrid Information Technology*, 2016.

[49] Angrosh Mandya, Danushka Bollegala, Frans Coenen, and Katie Atkinson. Combining long short term memory and convolutional neural network for cross-sentence n-ary relation extraction. *arXiv preprint arXiv:1811.00845*, 2018.

[50] Makoto Miwa and Mohit Bansal. End-to-end relation extraction using lstms on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116. Association for Computational Linguistics, 2016.

[51] Dat Quoc Nguyen and Karin Verspoor. Convolutional neural networks for chemical-disease relation extraction are improved with character-based word embeddings. *arXiv preprint arXiv:1805.10586*, 2018.

[52] Thien Huu Nguyen and Ralph Grishman. Combining neural networks and log-linear models to improve relation extraction. *CoRR*, abs/1511.05926, 2015.

[53] 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.

[54] Sampo Pyssalo, Filip Ginter, Hans Moen, Tapio Salakoski, and Sophia Ananiadou. Distributional semantics resources for biomedical text processing. In *Proceedings of the 5th International Symposium on Languages in Biology and Medicine, Tokyo, Japan*, pages 39–43, 2013.

[55] Pengda Qin, Weiran Xu, and Jun Guo. An empirical convolutional neural network approach for semantic relation classification. *Neurocomput.*, 190(C): 1–9, May 2016. ISSN 0925-2312. doi: 10.1016/j.neucom.2015.12.091. URL `https://doi.org/10.1016/j.neucom.2015.12.091`.

[56] Chanqin Quan, Lei Hua, Xiao Sun, and Wenjun Bai. Multichannel convolutional neural network for biological relation extraction. In *BioMed research international*, 2016.

[57] Deepak Ravichandran and Eduard Hovy. Learning surface text patterns for a question answering system. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 41–47. Association for Computational Linguistics, 2002.

[58] Nils Reimers and Iryna Gurevych. Optimal hyperparameters for deep lstm-networks for sequence labeling tasks. *arXiv preprint arXiv:1707.06799*, 2017.

[59] Nils Reimers and Iryna Gurevych. Reporting score distributions makes a difference: Performance study of lstm-networks for sequence tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 338–348, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1035. URL `https://www.aclweb.org/anthology/D17-1035`.

[60] Nils Reimers and Iryna Gurevych. Reporting score distributions makes a difference: Performance study of lstm-networks for sequence tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 338–348. Association for Computational Linguistics, 2017. doi: 10.18653/v1/D17-1035. URL `http://aclweb.org/anthology/D17-1035`.

[61] Bryan Rink, Sanda Harabagiu, and Kirk Roberts. Automatic extraction of relations between medical concepts in clinical texts. *Journal of the American Medical Informatics Association*, 18(5):594–600, 2011.

[62] Jonathan Rotsztejn, Nora Hollenstein, and Ce Zhang. Eth-ds3lab at semeval-2018 task 7: Effectively combining recurrent and convolutional neural networks for relation classification and extraction. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 689–696. Association for Computational Linguistics, 2018.

[63] Mohammed Saeed, Mauricio Villarroel, Andrew T Reisner, Gari Clifford, Li-Wei Lehman, George Moody, Thomas Heldt, Tin H Kyaw, Benjamin Moody, and Roger G Mark. Multiparameter intelligent monitoring in intensive care ii (mimic-ii): a public-access intensive care unit database. *Critical care medicine*, 39(5):952, 2011.

[64] Sunil Sahu, Ashish Anand, Krishnadev Oruganty, and Mahanandeeshwar Gattu. Relation extraction from clinical texts using domain invariant convolutional

neural network. In *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*, pages 206–215. Association for Computational Linguistics, 2016.

[65] Sunil Kumar Sahu and Ashish Anand. Drug-drug interaction extraction from biomedical texts using long short-term memory network. *Journal of Biomedical Informatics*, 86:15 – 24, 2018. ISSN 1532-0464.

[66] Cicero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. Classifying relations by ranking with convolutional neural networks. *arXiv preprint arXiv:1504.06580*, 2015.

[67] Isabel Segura-Bedmar, Paloma Martínez, and María Herrero Zazo. Semeval-2013 task 9: Extraction of drug-drug interactions from biomedical texts (ddiextraction 2013). In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, volume 2, pages 341–350, 2013.

[68] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

[69] Richard Socher, Brody Huval, Christopher D Manning, and Andrew Y Ng. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 1201–1211. Association for Computational Linguistics, 2012.

[70] Simon Suster, Madhumita Sushil, and Walter Daelemans. Revisiting neural relation classification in clinical notes with external information. In *Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis*, pages 22–28. Association for Computational Linguistics, 2018.

[71] Özlem Uzuner, Brett R South, Shuying Shen, and Scott L DuVall. 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association*, 18(5):552–556, 2011.

[72] Ngoc Thang Vu, Heike Adel, Pankaj Gupta, and Hinrich Schütze. Combining recurrent and convolutional neural networks for relation classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 534–539. Association for Computational Linguistics, 2016. doi: 10.18653/v1/N16-1065. URL `http://aclweb.org/anthology/N16-1065`.

[73] Linlin Wang, Zhu Cao, Gerard De Melo, and Zhiyuan Liu. Relation classification via multi-level attention CNNs. In *Proceedings of the 54th annual meeting of the Association for Computational Linguistics (volume 1: long papers)*, volume 1, pages 1298–1307, 2016.

[74] Peng Wang, Jiaming Xu, Bo Xu, Chenglin Liu, Heng Zhang, Fangyuan Wang, and Hongwei Hao. Semantic clustering and convolutional neural network for short text categorization. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, pages 352–357, 2015.

[75] Wei Wang, Xi Yang, Canqun Yang, Xiao-Wei Guo, Xiang Zhang, and Chengkun Wu. Dependency-based long short term memory network for drug-drug interaction extraction. *BMC Bioinformatics*, 18, 2017.

[76] Kun Xu, Yansong Feng, Songfang Huang, and Dongyan Zhao. Semantic relation classification via convolutional neural networks with simple negative sampling. *arXiv preprint arXiv:1506.07650*, 2015.

[77] Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. Classifying relations via long short term memory networks along shortest dependency paths. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1785–1794, 2015.

[78] Yan Xu, Ran Jia, Lili Mou, Ge Li, Yunchuan Chen, Yangyang Lu, and Zhi Jin. Improved relation classification by deep recurrent neural networks with data augmentation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1461–1470. The COLING 2016 Organizing Committee, 2016.

[79] Wenpeng Yin, Katharina Kann, Mo Yu, and Hinrich Schütze. Comparative study of cnn and rnn for natural language processing. *CoRR*, abs/1702.01923, 2017.

[80] Mo Yu, Matthew Gormley, and Mark Dredze. Factor-based compositional embedding models. In *NIPS Workshop on Learning Semantics*, pages 95–101, 2014.

[81] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344, 2014.

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

[83] Dongxu Zhang and Dong Wang. Relation classification via recurrent neural network. *arXiv preprint arXiv:1508.01006*, 2015.

[84] Shu Zhang, Dequan Zheng, Xinchen Hu, and Ming Yang. Bidirectional long short-term memory networks for relation classification. In *Proceedings of the*

*29th Pacific Asia Conference on Language, Information and Computation*, pages 73–78, 2015.

[85] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657, 2015.

[86] Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, 2017.

[87] Yuhao Zhang, Peng Qi, and Christopher D. Manning. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2018.

[88] Yuhao Zhang, Peng Qi, and Christopher D. Manning. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2205–2215. Association for Computational Linguistics, 2018.

[89] Zhehuan Zhao, Zhihao Yang, Ling Luo, Hongfei Lin, and Jian Wang. Drug drug interaction extraction from biomedical literature using syntax convolutional neural network. *Bioinformatics*, 32(22):3444–3453, 2016.

[90] Wei Zheng, Hongfei Lin, Ling Luo, Zhehuan Zhao, Zhengguang Li, Yijia Zhang, Zhihao Yang, and Jian Wang. An attention-based effective neural model for drug-drug interactions extraction. In *BMC Bioinformatics*, 2017.