University of Bonn

Fraunhofer IAIS

Master Thesis

Semantic Similarity in the Medical Domain

Author:
Abbas Goher Khan

Supervisor:
Prof. Dr. Stefan Wrobel

Second Supervisor:
Prof. Dr. Jens Lehmann

April 30, 2018

I hereby declare that this thesis was formulated by myself and that no sources or tools other than those cited were used.
Bonn, ...(Date) .....(Signature)

Acknowledgements

I would like to thank Sven Giesselbach and Dr. Stefan Rüping for their priceless contribution to the ideas and to the work on this thesis.

Abstract

Nowadays the amount of raw textual data potentially containing knowledge in various fields is extremely large. The necessity of abstract formal knowledge led to the task of Information Extraction and *Relation Extraction* as its part. This can be accomplished manually with the help of the experts in each of the considered domains, but of course, experts' time and knowledge are expensive and limited.

All aforementioned gave rise to the attempts of solving the problem and delivering knowledge by automated machine learning methods. This solution still requires experts' skills, but less than in the case of manual knowledge extraction. One of the promising directions for working on Relation Extraction is *Deep Learning*. But Deep Learning models usually require a lot of training data in order to learn. Taking into account that best manual labels are hard to get the approach of *Distant Supervision* was used in this thesis. Distant Supervision allows using a small amount of gold standard data in order to get a large amount of approximate training data.

The *medical domain* can be seen as an example field. It is very critical to have automated knowledge extractors for publications and articles that might contain the answers to everyday questions in doctoral practice. Taking into account the large volume of research and a few experts in the area, the domain creates a perfect example of the domain that needs automating of the knowledge extraction as developed in this thesis.

The goal of this thesis is to investigate the possibilities of Convolutional Neural Networks together with Distant Supervision and Multiple Instance Learning in solving the problem of Relation Classification. In order to see the power of the approach, it is tested not only in the medical domain but in a general domain as well. The approach of supervised training is compared to Distant Supervision to find out the benefits and drawbacks of the latter. Multi-Instance Learning is considered as an approach that can improve Distant Supervision.

Overall experiments proved the possibility of using existing knowledge and raw textual data for automatically created training dataset that allows training the model to sufficient level without manual labelling. Distantly supervised data added to existing supervised data might improve Recall, that allows experts to concentrate attention only on classified sentences and not to check all additional textual data.

Contents

1 Introduction **4**
- 1.1 Motivation . . . 4
- 1.2 Goals . . . 6
- 1.3 Structure of the thesis . . . 7

2 Background **8**
- 2.1 Summary of relevant approaches . . . 8
  - 2.1.1 Word and Document Embeddings . . . 10

3 Approach **13**
- 3.1 Model description . . . 13
  - 3.1.1 Ranking convolutional neural network . . . 14
  - 3.1.2 Objective . . . 16
  - 3.1.3 Training . . . 18
- 3.2 Distant supervision . . . 19
  - 3.2.1 Multiple instance learning . . . 20
- 3.3 Interpretability evaluation . . . 21
  - 3.3.1 Representative trigrams . . . 21
  - 3.3.2 Semantic values . . . 22
  - 3.3.3 Scores distribution . . . 23
- 3.4 Implementation details . . . 23

4 Experiments **27**
- 4.1 Supervised training evaluation, general domain . . . 28
  - 4.1.1 SemEval 2010, Task 8 . . . 28
  - 4.1.2 KBP37 . . . 31
  - 4.1.3 Results . . . 32
  - 4.1.4 Interpretation . . . 34
- 4.2 Supervised training evaluation, medical domain . . . 39
  - 4.2.1 AIMed . . . 40
  - 4.2.2 DDI . . . 41

4.2.3 Rosario-Hearst dataset 42
4.2.4 Results 42
4.2.5 Interpretation 45
4.3 Distant supervision evaluation 48
4.3.1 General domain: KBP37 based distantly supervised dataset 48
4.3.2 Results 50
4.3.3 Interpretation 51
4.3.4 Medical domain: Rosario-Hearst based distantly supervised dataset 55
4.3.5 Results 56

5 Conclusions **59**
5.1 Conclusions 59
5.2 Future work 60

List of figures **62**

List of tables **63**

References **64**

Chapter 1

Introduction

Automated knowledge discovery methods are extremely important nowadays because of the amount of unstructured textual data. In order to utilise the information contained in the raw texts, one needs to find all the various objects mentioned there and find what kind of connections are between them according to the text. This is a quite demanding and time-consuming problem that requires experts' knowledge and skills. Thus the goal is to solve it by application of machine learning method, specifically Artificial Neural Network trained in a way that requires as less as possible experts' work.

1.1 Motivation

Knowledge Bases and Ontologies are crucial parts of any of expert system that are designed to help professionals in their work. In order to obtain such a Knowledge Base, domain experts have to invest a lot of their costly time. They also have to curate the Knowledge Base over and over again because new knowledge is acquired and published continuously.

Usually, the subjects of interest are people, locations or organisations that are considered as *entities*. Knowledge extraction is defined as the comprehension of the semantic meaning behind textual data, i.e. understanding which kinds of known relations connect entities. Most common relations are binary, e.g. employee-of(person, organisation). If we think about more specific domains then entities might be genes, proteins, diseases for the biomedical domain, or algorithms, concepts and applications for the computer science domain and so on. Certainly, relational classes can also become more specific.

Possible applications of the extracted knowledge are very diverse. One prototypical application is an expert system for collecting knowledge in the domain. For example in the Electronic Patient Path project, a system is

aiming to help medical workers to find individual therapies for treating colorectal cancer. It has to contain a lot of knowledge about the field in order to give adequate answers to given questions. Its core is a Knowledge Base containing all related entities and relations between them. Manipulation of such an up to date knowledge allows to quickly understand how to treat a new patient.

In order to better understand the idea behind Information Extraction, consider the following simple example.
"Interior Minister Giuliano Amato told lawmakers on Tuesday that more arrests were likely in Rome after police identify rioters. Holy Cross High School is a Catholic secondary school founded in Waterbury Connecticut in 1968 by the Congregation of Holy Cross. De La Salle High School was founded by the Christian Brothers. "
The text above contains several mentions of different people and organisations. Obviously, it should also contain some information about their interactions. But just after reading the sentences time and effort are required to answer questions like:
"Who is Giuliano Amato? When and by whom was founded Hole Cross High School? Who founded De La Salle High School?"
The task of information extraction is to simplify this process. Specifically, the process consists of finding entities or entity mentions:
"Interior Minister, Giuliano Amato, Rome; Holy Cross High School, Waterbuty, 1968, Congregation of Holy Cross; De La Salle High School, the Christian Brothers"
and then interpreting the semantic meaning behind the text connecting these entities. Ideally, information extracted from this text will look like diagram depicted in the Figure 1.1. Such kinds of diagrams can be seen as Knowledge Graphs and usually they are the most popular way of displaying content of a Knowledge Base. Having such a Knowledge Base answering aforementioned questions might be completely automatised thus requiring no human effort. It should be noticed, that sometimes the text will not contain enough information to name the exact relation between some of the marked entities. For example, the considered sentence does not contain any exact relation between *Giuliano Amato* and *Rome*. This is important, as marking all the known entities in the textual data will create a lot of noise that should be somehow filtered during the Relation Extraction process.

For now, the creation of a Knowledge Base is still a problem that requires experts work. Automated methods currently suffer from many pitfalls, e.g exploiting pipelines of Natural Language Processing tools which are not always accurate and introduces its own miscalculations that lead to the higher overall error rate. Another challenge is the necessity of training examples for

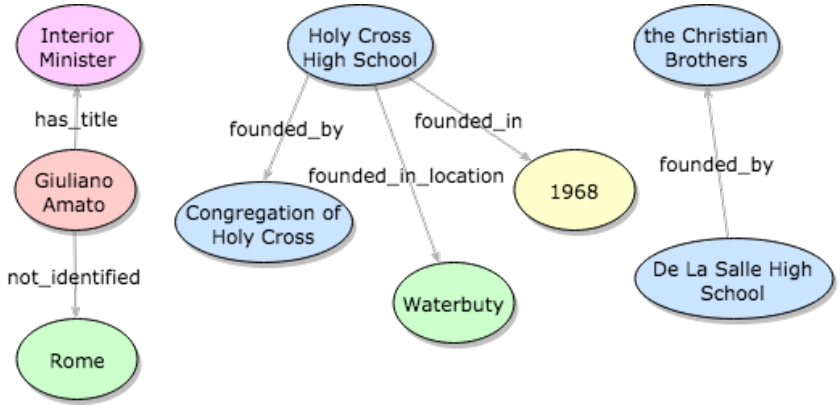


Figure 1.1: Knowledge extracted from textual data in the form of Knowledge Graph.

most of the machine learning models. There usually does not exist sufficient amount of labeled examples of relations in natural texts because labelling texts is a very time-consuming and effort demanding activity.

Nevertheless, there is a huge volume of raw textual data and some already created and curated Knowledge Bases that definitely can be very helpful in solving the problem of Information Extraction. The process of knowledge discovery might be improved and all existing structured and unstructured data should be exploited for it.

1.2 Goals

The main idea in this work is to explore the ways in which deep-learning can be used for solving the problem of Relation Extraction and Classification. This research plans to tackle this problem in general and for the medical domain in particular.

The goal is to improve Relation Extraction by usage of Deep Learning methods together with weak supervision and Multiple Instance Learning. The chosen approach is to apply the Convolutional Neural Network proposed in [dos Santos et al., 2015] combined with Distant Supervision [Mintz et al., 2009] and Multiple Instance Learning [Zeng et al., 2015]. The idea behind is that a Convolutional Neural Network should be able to identify critical parts of a sentence and transform them to a feature vector that can be identified as belonging to one or other relation. The concept of the Distant Supervision aims to remove the need of manually labeled examples or at least limit the needed amount. It exploits the knowledge contained in the existing

sources of structured data and makes use of raw unstructured texts aligning them together for getting training datasets. As it suffers from very general assumption that
''Every sentence containing two related entities will describe this relation"
a lot of methods for improving it were studied. One of them is Multiple Instance Learning that mitigates the assumption by giving a label not to one sentence, but to the set or so-called bag of sentences with the same entities pair and saying that at least one of them should describe this relation.

One of the questions that can be answered is how Distant Supervision affects the Relation Extraction problem solved with Convolutional Neural Network. It is important to understand how and where experts should be involved in order to obtain a maximal result with minimal effort and time.

1.3 Structure of the thesis

Experiments are aimed to answer questions about the helpfulness of the Distant Supervision approach and the possibilities to improve knowledge discovery with minimally manned methods. Any textual information available can be considered as a source of examples for the general domain and there exist various public Knowledge Bases. As it is easier to validate the result in general domain than in specific medical one, the first part of experiments are performed for general domain data and then the model is applied to the medical domain. For the medical domain a large volume of raw texts from PubMed [1] along with existing medical Knowledge Base and ontologies is used.

The structure of the text is following:

- Chapter 2 describes the background knowledge about the problem of Natural Language Processing, Relation Extraction and lists relevant approaches
- Chapter 3 specifies the exact approach chosen for this thesis and describes various implementation details
- Chapter 4 shows evaluation results on different datasets in different domains
- Chapter 5 suggests future work and observations made during performing the evaluation of the approach

[1] `https://www.ncbi.nlm.nih.gov/pubmed`

Chapter 2

Background

Information retrieval (IR) is the activity of obtaining information resources which are relevant to a given query. In terms of semantic similarity, the task of information retrieval is to find documents which are semantically similar to a given query. One of the easiest approaches for finding relevant documents given a query is TFIDF. TFIDF, short for term frequency inverse document frequency, is a information retrieval technique that shows how important a word is to a document. Another information retrieval algorithm known as BM25 (BM stands for Best Match) [Robertson et al., 2009] can be used to retrieve matching documents according to their relevance to a given query. [Dumais, 2004] introduced a word embedding method as an extension of TDIDF known as Latent Semantic Analysis (LSA). [Mikolov et al., 2013a], [Pennington et al., 2014] and [Shazeer et al., 2016a] have introduced neural network based word embedding models which have now become benchmarks in semantic similarity extraction.

2.1 Summary of relevant approaches

The most naive and intuitive way of finding relevant documents given a query term is to use term frequency (TF). TF assumes that the more frequently the given query term appears in a document the more relevant that document is to the given query. TF however suffers from a critical problem: all words are considered equally important when it comes to assessing relevancy on a query. In fact, though some words appear multiple times in a document they have very little discriminating power in determining the relevancy. For instance, a collection of documents on the football is likely to have the word football or soccer in almost every document. To circumvent this issue a technique known as inverse document frequency (IDF) is used

to attenuate the effect of words that occur too often in the collection of documents to be meaningful for relevance determination.

We define inverse frequency of a word w as follows:

$$\mathrm{IDF}_w = \log \frac{N}{\mathrm{DF}_w}.$$

Where DF_w is the document frequency and is defined as the number of documents in the collection that contain a word w and N is the total number of documents. Thus the IDF of a rare word is high, whereas the IDF of a frequent word is likely to be low. Each word has its own TF and IDF score, the product of the two scores is called the TFIDF weight of that word. [Ramos et al., 2003] provide evidence that TFIDF returns documents highly correlate to the given query. Different weight schemes for these counts lead to a variety of TFIDF ranking features. One very successful TFIDF formulation is known as BM25 ([Robertson et al., 2009]). BM25 is a bag-of-words information retrieval function that ranks a set of documents based on their relevance to the query terms. Tweaking different components and parameters produce different variations of the BM25. [Mitra et al., 2016] have shown BM25 to be effective in information retrieval and have also proposed using it in an ensemble model along with other embedding model, namely Word2Vec.

The TFIDF vectors tend be large since they have one component for every word in the vocabulary. [Berger et al., 2000] propse a number of extensions to TFIDF, including what they call Adaptive TFIDF. This algorithm incorporates hill-climbing and gradient descent to improve the performance.

Another extension introduced by [Dumais, 2004] is known as Latent semantic analysis LSA. LSA uses Singular Value Decomposition (SVD) to perform dimensionality reduction on the TFIDF vectors resulting in smaller and better features. In LSA documents are represented as bags-of-words, where the order of the words in a document is not important, only how many times each word appears in a document. Furthermore, LSA assumes that words which are close in meaning will appear in similar pieces of text, A concept known as distributional hypothesis. [Boling and Das, 2014] has shown that LSA performs very well in finding semantic similarity between documents. Other popular models which use distributional hypothesis are Word2Vec ([Mikolov et al., 2013a]) and GloVe ([Pennington et al., 2014]).

2.1.1 Word and Document Embeddings

Word embedding is a language modeling and feature learning technique in natural language processing(NLP) which maps words and phrases to the real number vector space of desired dimension.

Word2Vec Word2Vec model was introduced by [Mikolov et al., 2013a]. It uses distributed vector representation of words, a well-known framework for learning word vectors as shown in the Figure 2.1. The task is to learn to predict a word given other words in the context. More formally, given a sequence of training words $w_1, w_2, w_3, ..., w_T$, the objective of the word vector model is to maximize the average log probability

$$\frac{1}{T}\sum_{t=K}^{T-K} \log p(w_t \mid w_{t-1}, ..., w_{t+1}) \tag{2.1}$$

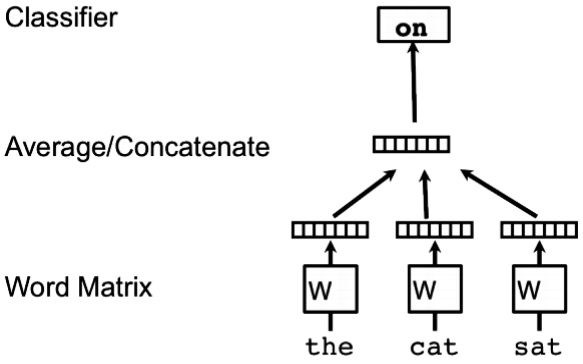


Figure 2.1: A framework for learning word vectors. Context of three words (the, cat, and sat) is used to predict the fourth word (on). The input words are mapped to columns of the matrix W to predict the output word.

[Bojanowski et al., 2016] propose another extention of Word2Vec model known as FastText. It learns word representations while taking into account morphology. FastText models morphology by considering subword units, and representing words by a sum of its character n-grams. Since FastText exploits subword information, It can also compute valid representations for out-of-vocabulary words. FastText obtains representations for out-of-vocabulary words by summing the vectors of character n-grams.

[Ghosh et al., 2016] introduce a vocabulary driven Word2Vec method known as Dis2Vec which is used to generate disease specific word embeddings from unstructured health related news corpus. The input corpus D consists

of a collection of word context pairs. Based on the vocabulary V, we can categorize the word context pairs into three types as shown below:

- $D(d) = (w, c) : w \in Vc \in V$, i.e. both the word w and the context c are in V
- $D(\rightharpoonup d) = (w, c) : w \notin Vc \notin V$, i.e. neither the word w nor the context c are in V
- $D(d)(\rightharpoonup d) = (w, c) : w \in V \oplus c \in V$, i.e. either the word w is in V or the context c is in V but both cannot be in V

Each of these categories of (w, c) pairs needs special consideration while generating disease specific embeddings.

All the above mentioned word embedding models learn word embeddings from co-occurrence information in corpora. One drawback of learning word embeddings by by this approach is that such methods will generally fail to tell synonyms from antonyms ([Mohammad et al., 2008]). For example, words like east and west or expensive and inexpensive appear in near-identical contexts, which means that distributional models produce very similar word vectors for such words. Such embedding are very undesirable when the goal is to find semantic similarity between documents. [Mrkšić et al., 2016] proposed a novel counter-fitting method which injects antonym and synonymy constraints into vector space representations in order to circumvent this issue. Table 2.1 shows the results [Mrkšić et al., 2016] achieved using their counter-fitting technique.

An alternative to the bag-of-words approach is to derive contexts based on the syntactic relations the word participates in as proposed by [Levy and Goldberg, 2014].

All the above mentioned approaches are word embedding models and do not generalize to sentences and documents. [Jśnior et al., 2017] propose two methods for obtaining sentence and document level embeddings. The first approach obtains vector embeddings for documents by averaging the word embeddings of all the words in a document. The second approach also averages word embeddings, but each embedding vector is now weighted (multiplied) by the TFIDF of the word it represents.

Doc2Vec An extension to the Word2Vec model known as Doc2Vec was introduced by [Le and Mikolov, 2014]. Doc2Vec is capable of constructing representations of input sequences of variable length. Unlike some of the

	east	expensive	British
Before	west	pricey	American
	north	cheaper	Australian
	south	costly	Britain
	southeast	overpriced	European
	northeast	inexpensive	England
After	eastward	costly	Brits
	eastern	pricy	London
	easterly	overpriced	BBC
	-	pricey	UK
	-	afford	Britain

Table 2.1: Nearest neighbours for target words using GloVe vectors before and after counter-fitting

previous approaches, it is general and applicable to texts of any length: sentences, paragraphs, and documents. In Doc2Vec framework (see Figure 2.2), every document is mapped to a unique vector, and every word is also mapped to a unique vector. The document vector and word vectors are averaged or concatenated to predict the next word in a context. The only difference to a Word2Vec model is the additional document token.It acts as a memory that remembers what is missing from the current context or the topic of the document. The document vectors and word vectors are trained using stochastic gradient descent and the gradient is obtained via back-propagation.

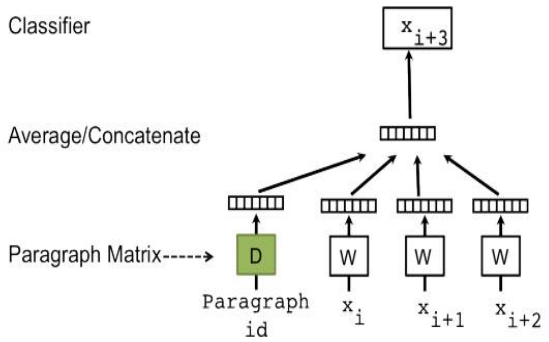


Figure 2.2: A framework for learning paragraph vector. This framework is similar to the framework presented in Figure 1; the only change is the additional paragraph token that is mapped to a vector via matrix D.

Chapter 3

Approach

Based on the background introduced in Chapter 2, a concept of a Convolutional Neural Network model, similar to the one proposed in [dos Santos et al., 2015], but trained by means of Distant Supervision and improved by Multiple Instance Learning was developed. An approach for implementation and evaluation on the selected datasets was introduced. First of all, the selected neural model should be implemented with one of the present programming libraries. The implementation should be proven to be correct via comparing the results achieved in the paper describing the model. Also, it is always useful to understand the underlying mechanism of the work of the network, thus different methods for interpretation of results might be utilised. The next main step is to make use of available Knowledge Bases and textual corpora by applying Distant Supervision. Afterwards, various methods for improvement of the achieved results can be implemented.

3.1 Model description

The model implemented for the research is a Convolutional Neural Network, that uses the idea of ranking for classification. It was implemented from scratch with Python, Tensorflow [1] and Keras [2] libraries. As the idea of application of Deep Learning is to replace hand-crafted features for classifying relations Convolutional Neural Network model was selected instead of Recurrent Neural Network. Among all the models introduced for today the results of [dos Santos et al., 2015] showed up to be most impressive - so in [Nguyen and Grishman, 2015] a combination of Recurrent Neural Networks and Convolutional Neural Networks gives the same results, in [Vu et al., 2016]

[1] https://www.tensorflow.org/
[2] https://keras.io/

	GloVe [3]	Word2Vec [4]	Swivel [5]
300	downloaded, Wikipedia dump from 2014 and Gigaword 5	downloaded, Google News corpus	trained locally, Wikipedia dump from 2016
400	trained locally, Wikipedia dump from 2016	trained locally, Wikipedia dump from 2016	trained locally, Wikipedia dump from 2016

Table 3.1: Summary of the embeddings used for the experiments.

a combination just slightly improves the result and a model based on various Natural Language Processing features [Gormley et al., 2015] also performs worse. And evaluation later showed that simple Recurrent Neural Network performed worse on one the testing datasets.

3.1.1 Ranking convolutional neural network

The model has the following structure:

1. **Words embeddings layer** The layer is responsible for transforming words of the input sentence to the embeddings. Every word w_i of the sentence is transformed to a vector r^{w_i}. Every such vector is a row of an embedding matrix W^{wrd} for some fixed-size vocabulary. Three types of word embeddings were used for experiments, in order to compare their effectiveness for the concrete task: Word2Vec [Mikolov et al., 2013b], GloVe [Pennington et al., 2014] and Swivel [Shazeer et al., 2016b]. For each type pre-trained versions were used if they were available on the internet. This was done because, as a rule, the quality of such versions is comparatively high. Thus there is a variety of corpora that embeddings were trained on. Information about the used embeddings can be found in the Table 3.1. The evaluation showed that downloaded embeddings are more effective on average, thus proving the point that accurate tuning of parameters and cross-validation on various tests of quality for word embeddings can improve the result of the final model as well.

[3] Implementation and pre trained embeddings are from `http://nlp.stanford.edu/projects/glove/`

[4] Implementation from `https://github.com/RaRe-Technologies/gensim`, pre trained embeddings from `https://drive.google.com/file/d/`

The size of word embeddings d^w is obtained by 4-fold validation experiments. Embeddings are modifiable during the training of the network, as it gives better results than keeping them constant (proved empirically in [dos Santos et al., 2015]).

2. **Distance embeddings layer** The layer is responsible for transforming distances between the words in the sentence and marked named entities to the embedding vectors wp_1 and wp_2. Essentially, the vocabulary of these embeddings is just numbers in the range $[p-l, p+l]$ where l is the length of the longest sentence in the dataset and p is some arbitrary big number, larger than the length of the longest sentence. The idea behind these embeddings is to point up the entities that are checked for the relation to the network. This approach was introduced by [Zeng et al., 2014] and widely used in many other works. There are also other ways of stressing the entities, for example taking into account only part of the sentence between them, but according to the evaluation in [dos Santos et al., 2015] position embeddings give the best results. The size of embeddings d^{wpe} is obtained by 4-fold validation experiments. The grid for search consisted of 30, 40, 50 and 70, as 70 was the embedding size used in the [dos Santos et al., 2015] and the intuition behind embeddings is that for smaller vocabulary smaller dimension of embedding is needed. Embeddings are initialized with random numbers uniformly distributed in 0,1 and learned during training of the network.

3. **Embeddings merge layer** The layer concatenates the word embedding r^w and corresponding distance embeddings (to the first named entity and to the second named entity) wp_1 and wp_2 for every word w in the input sentence into one vector.

4. **Convolutional layer** Convolution is applied to windows of three embedding vectors with zero padding, so the size of the input is not changed after the layer application. The number of filters d^c is 1000 [dos Santos et al., 2015], i.e. application of the filters results into 1000 vectors of the same length as the sentence, where each value in one vector is a feature value for a specific triplet of words. The activation function applied after the filtering is *tanh* [dos Santos et al., 2015].

5. **Global max pooling** The maximal value is found in the output of each filter. After this layer network has formed the universal representa-

`0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit`

[5]Implementation from `https://github.com/tensorflow/models/tree/master/swivel`

tion of any sentence r_x that is a real-valued vector in 1000-dimensional space. Application of global maximum allows not to take into account different lengths of input sentences.

6. **Scoring dense layer** In order to classify relations the closeness of a sentence representation to real-valued vectors representing each of the relations that are learned during the training process is estimated. I.e. dot product of a sentence representation r_x and each of the vectors that are called relation embeddings is calculated. The embedding that yields the maximal score is considered to be the embedding of the relation described in the sentence. The output of the network is the array of scores, for each of the classifiable relations. The scoring procedure is implemented as a dense layer without bias with weights matrix $W^{classes}$ consisting of the relations embeddings. The weights are initialised randomly uniformly in the interval

$$\sqrt{\frac{6}{|C| + d^c}}$$

where C is a number of classifiable relations, d^c is the length of sentence representation [dos Santos et al., 2015].

3.1.2 Objective

The objective function uses two of the resulting scores - one is the score, that was obtained for the correct relation according to the label of the example and the second score is one of the wrong relations scores. Ideally, according to [dos Santos et al., 2015] the second score should be the maximal score given to the wrong relations.

Thus, objective calculation consists of the following steps:

- Get the score to the correct relation from the array of scores and the correct label;
- Get the maximal score from the remaining wrong relations;
- Calculate the value of the loss according to the formula:

$$L = log(1 + exp(\gamma(m^+ - s_\theta(x)^+_y))) + log(1 + exp(\gamma(m^- + s_\theta(x)^-_c)))$$

where m^+ is a margin for the right answer; m^- is a margin for the wrong answers; γ is a scaling factor; $s_\theta(x)^+_y$ is a score for the right

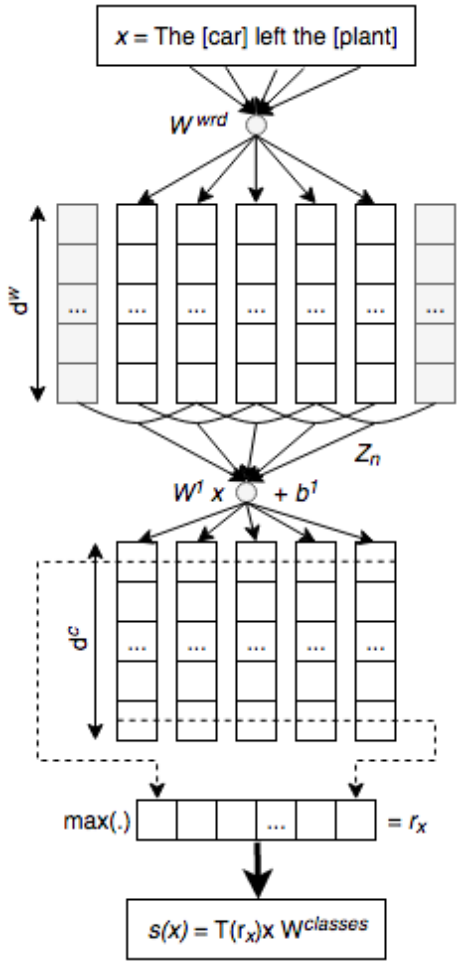


Figure 3.1: Convolutional Neural Network for Relation Extraction with ranking [dos Santos et al., 2015]. The input to the network is a sentence with marked entities. The first transformation is performed by the matrix of word embeddings W^{wrd} to obtain a dense representation of each word. Embeddings of the distances to the entities are obtained and concatenated together with the word embeddings. The second step is a convolution that results in d^c values of filters for each window. Then global max pooling is applied to obtain the final representation of the sentence r_x that is compared to each of the relation embeddings in the weight matrix $W^{classes}$.

class; $s_\theta(x)_c^-$ is a score for the wrong class. Margins and scaling factor are fixed numbers and their values are $m^+ = 2.5$, $m^- = 0.5$, $\gamma = 2$ as in [dos Santos et al., 2015].

Intuitively, the minimisation of the loss leads to increasing the gap between the wrong score and the right score by minimisation of the first one and maximisation of the second one.

One special case for the objective function is behaviour with the class "Other". Class "Other" includes all the examples that are either describing relations that are not among main classes or do not describe any relation at all. According to [dos Santos et al., 2015] trying to make the network learn the embedding for the class "Other" leads to worse results, as it is very noisy and might also affect the results obtained for other classes. Therefore, when an example of "Other" class shown to the network objective function changes the first term (containing score for the right class) to zero. In this way the network is not learning anything about the "Other" class. During the recognition process example will be classified as "Other" only when the scores for all classes are negative. I.e. only when the network cannot classify example as any of desired classes it will reply that the class is unknown.

The loss also includes regularisation. L2 regularisation is applied on weights of the convolutional layer and the embeddings of the relations (i.e. the weights of the last dense layer) with the regularisation rate 0.001 [dos Santos et al., 2015].

A more common way for multi-class classification is an application of a softmax classifier after a dense layer to obtain the probability distribution. But according to evaluation experiments in [dos Santos et al., 2015] ranking performs at least two percents better than softmax.

3.1.3 Training

Training is done by simple gradient descent with decaying learning rate. The learning rate is divided by the number of the epoch starting with the initial value of 0.025 [dos Santos et al., 2015]. Usually, the network almost stops learning after 10-20 epochs, because afterwards the gradient updates become very small and almost do not affect the result. The validation accuracy also stops improving and keeps fluctuating around the value reached before.

The training is organised by giving one sentence at a time, so the training data does not need to be padded to the same length. Epoch finishes after showing the network all sentences in the training set. After each epoch, the training set is shuffled randomly.

3.2 Distant supervision

As discussed in Chapter 2, Distant Supervision is used in order to overcome the problem of scarcity of training data for the neural network. In order to apply this concept, a Knowledge Base that contains desired relations is required along with large text corpus. An existing Knowledge Base is used for generating examples of relation mentions. This is achieved by aligning the entities from the Knowledge Base with the text corpus, either by simple string matching or more complex entity recognition solutions. Each sentence containing entity pair mention is considered to be labeled with the relation known for this pair from the Knowledge Base. So there are two assumptions made here:

1. For every triple (e_1, e_2, r) in a Knowledge Base every sentence containing mentions for e_1 and e_2 express the relation r

2. Every triple that is not in a Knowledge Base assumed to be a false example for relation, while the reason might be in the incompleteness of knowledge.

During the construction of distantly supervised training dataset, it is important to take into account the number of examples for each of the entity pairs and for each of the relations. It is very common that some of the pairs are much more often mentioned compared to other ones and the same is valid for the amounts of sentences for each relation because a Knowledge Base will not be balanced by the amount of triples for different relations.

There are different ways to evaluate the result of a distantly supervised model, one of them, for example, is to choose supervised dataset with the same relational classes and test the model. The main challenge in this approach is to find correspondence between existing labeled datasets and a public Knowledge Base. Empirically, it was found out that relations should be maximally similar, otherwise distant data will only introduce noise to the training process. Also, the corpora for aligning affect the quality of distantly supervised dataset a lot. [Riedel et al., 2010] compared the amount of wrongly labeled examples when aligning Freebase to Wikipedia corpora and to New York Times news corpora. As it was expected Wikipedia has usually almost 10% fewer mistakes. For example, if the relation "Nationality" is considered, it is quite possible that on news the country will be mentioned together with the person, but the exact relation of nationality is not described there. While Wikipedia will definitely contain sentences about the country of birth and residence of the person.

One more important point that should be taken into account when validating distantly supervised model with the existing testing dataset is that the underlying distribution behind supervised dataset most probably will differ from the distribution obtained from textual corpora [Craven et al., 1999]. Same distribution in both training and testing datasets is one of the base assumptions for any supervised learner and it is extremely hard to control with distantly supervised datasets taking into account that even approximately same distribution in classes will not guarantee the same distribution of true underlying features of relations.

In [Craven et al., 1999] it is claimed that distantly supervised data improves precision but on the other hand following the results from [Riedel et al., 2010] it also introduces noise, that definitely does not improve the precision. But on the other hand text corpora are very vast and definitely contain much more various syntactic constructions for defining specific relation, thus it might improve the recall of the model.

So several notes should be of importance for Distant Supervision:

1. Exact correspondence between relations in Knowledge Base and desirable or described in the supervised dataset;

2. Appropriate textual corpora that contain information about desirable relations;

3. Sufficient variety of syntactic constructions in testing set for adequate evaluation of the results.

3.2.1 Multiple instance learning

The assumption of the Distant Supervision is very imprecise as mostly sentences do not describe the relation that is stored in the Knowledge Base exactly. For solving this problem Multiple Instance Learning might be applied [Zeng et al., 2015].

This approach was first introduced for drugs classification - whether they are active or not [Dietterich et al., 1997]. The problem over there that every drug contains different molecules that are active or not, but only the activity of the drug is known. So in order to learn information about activity Multiple Instance Learning is performed - one label for several instances is known. One simple example can be seen in the Figure 3.2.

In the application for Relation Extraction Multiple Instance Learning will mean, that we assume the existence of at least one sentence containing the description of the relation from the Knowledge Base. One of the possibilities to construct bags is to unite the sentences with same entities mentions in

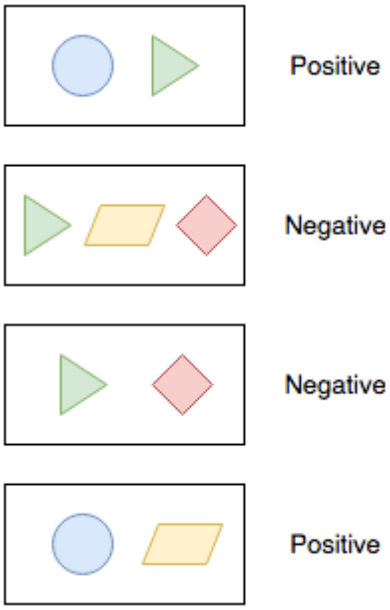


Figure 3.2: Example of a dataset for Multiple Instance Learning. One of the hypotheses that can be suggested is that a blue circle makes the label positive, so this is the most important feature of a 'bag'.

one bag and give a corresponding label of the relation from the Knowledge Base. There are also different ways to train a neural network with bags. The way from [Ramon and De Raedt, 2000] was chosen, i.e. the maximal score example should be chosen from the bag every time to fit the model. And all the bags are shuffled from epoch to epoch.

This approach is very naive and looses a lot of possibly useful information obtained by Distant Supervision, but it still can be used as an initial step for possible improvement of the approach.

3.3 Interpretability evaluation

In order to better understand the work of the neural network and the concept that was learned various insights could be used. Several of them were applied in this work.

3.3.1 Representative trigrams

According to [dos Santos et al., 2015] so-called representative trigrams can be extracted for classifiable relations. A representative trigram is a three

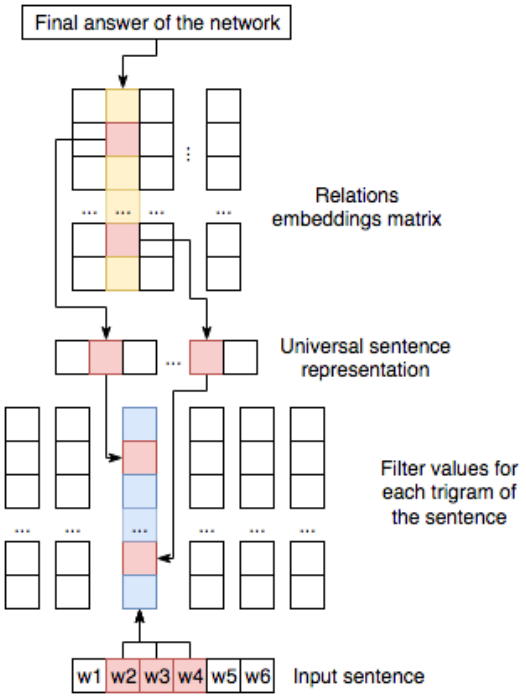


Figure 3.3: Schematic view of tracing back the trigram, that made corresponding input to the final score of the relational class.

words combination, that characterises best (according to the network) one of the classes. They can be obtained from the sentences of the dataset by measuring the value that each of trigrams in a sentence adds to the correct class score. The value can be simply seen as a score that is obtained for the relational class, if only one trigram of the sentence is seen. For better understanding of the way to obtain the score for the trigram the Figure 3.3 can be seen.

This method is very similar to the one mentioned in [Craven et al., 1999] when the most valuable words were extracted in order to have an insight into the concept learned by the model. For example, if "Origin" relation is learned, one can expect to see the trigrams *was born in, country of birth, was originated in*, etc.

3.3.2 Semantic values

The approach of [Zhang and Wang, 2015] can be used to measure semantic values of each word in a sentence for classifying a relation between entities in the sentence. The idea of calculating is very close to finding representa-

tive trigrams, but in this case, the number of components of the resulting 1000-dimensional description of a sentence is taken as a characterising value. Thus, the initial trigram from the sentence was traced back for each component of a sentence description. The number of components of the sentence description vector traced back to the trigram is normalised by the dimension of the description vector (1000). This value is thought of as a semantic value for the central word of the trigram. There is a strong connection between representative trigrams and semantic values of the words of the sentence, as one can expect that the more characterising the words are, the larger their semantic value. For example, in the sentence
"$<$ $e1$ $>$George Walker Bush$<$ $/e1$ $>$ was born on July 6, 1946, at Grace-New Haven Hospital (now Yale?New Haven Hospital) in New Haven, $<$ $e2$ $>$Connecticut$<$ $/e2$ $>$, as the first child of George Herbert Walker Bush and his wife, the former Barbara Pierce." (*Origin*)
one can expect large semantic values for words *was born*.

3.3.3 Scores distribution

The way to understand better the answer given by the network is to display the scores distribution across the classes for a sentence. It can provide with information about how sure is the network about its answer that is usually an important characteristic of the machine learning model. Thus, big difference of scores between the right relation and wrong relation can show, that the network is sure and trained well enough for recognising this relation.

3.4 Implementation details

The network was implemented using Tensorflow, a Python library for tensor calculations and the high-level library Keras, that allows making the code more readable and understandable.

The experiments were run on NVIDIA Corporation GM200 [GeForce GTX TITAN X].

The structure of the network was visualised with Tensorboard [6] (Figure 3.4) that allows checking the correspondence to the desired structure.

Following are several implementation details are listed:

- The input to the network is a single sentence with two marked named entities. Before application of convolution preprocessing is done. Preprocessing includes tokenisation and calculation of distances of each

[6] `https://www.tensorflow.org/get_started/summaries_and_tensorboard`

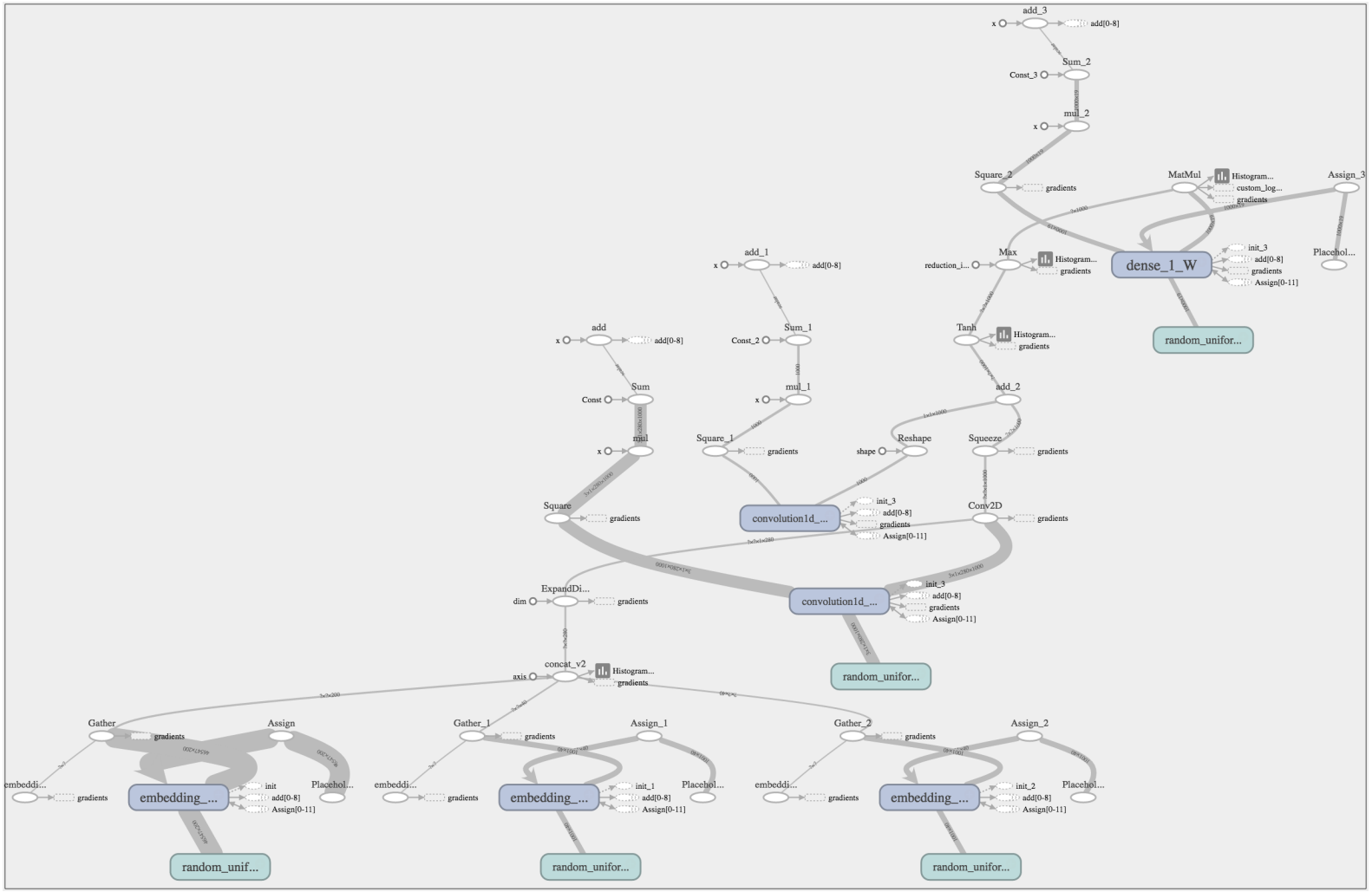


Figure 3.4: Screenshot from tensorboard visualisation of the implemented network. One can see three merging embedding layers in the bottom, that are followed by the convolutional layer. Then application of *tanh* follows, after which *Max* function and dense layer to get scores for each of the relations.

word to the first and second named entities. Distance here is simply a number of words in between. It is negative when counted to the left and positive when counted to the right. For example, in the sentence "The *car* left the *plant*." for token "left" distance to the first entity "car" is -1 and to the second entity "plant" is 2. The tokenisation was carried out by Penn Treebank tokeniser [7] that is delivered by NLTK python package [8]. The only modification that was applied on the text before tokenisation is down casing. The first layers of the network are embedding layers. They are applied in order to transfer tokens and distances to embedding vectors. Essentially, the task of an embedding layer is to substitute current token with its embedding, known beforehand. Usually, embeddings are stored in the form of the matrix, so the most straightforward implementation of an embedding layer is to find correspondence between a token and a row in the matrix. Thus all the tokens are transformed to indices, that represents the number of the row with embedding for this token. This functionality causes a problem for distance embeddings as distances might be negative when counted to the left of the token and cannot be used directly as indices in embeddings matrix. In order to solve this issue, each distance was made positive by adding a large positive number p. As the maximal negative number cannot be larger by absolute value than the length of the longest sentence, it is easy to calculate this large number p.

- In order to properly work with embeddings, the whole vocabulary of the embeddings should be used in the matrix for the embeddings layer. But due to the Keras implementation of the training, it becomes extremely slow with large sizes of the weights matrix. Also not all the embeddings are supposed to change on each step. Therefore, it was decided to limit the vocabulary only to the words that are there in the training and testing dataset. Also, Keras implements dense updates for embedding layer, that extremely slows down the training, while Tensorflow implementation uses sparse update. In order to get faster training native Tensorflow Gradient Update was used [9].

- Also, the vocabulary for the distances between words and entities was not described in the original paper. It was chosen in the following way. Each distance was summed with a large number p (greater than the

[7] `https://nlp.stanford.edu/nlp/javadoc/javanlp/edu/stanford/nlp/process/PTBTokenizer.html`

[8] http://nltk.org

[9] https://github.com/fchollet/keras/issues/4365

length of the longest example) in order to make it positive and use as an index in embeddings matrix. Thus, vocabulary consists of all the numbers from 0 to $2 * p + 1$.

- As in the original paper sentence padding was not mentioned, the examples were decided to give one-by-one and gradient step was performed after each sentence.

- The output of the network also was not described precisely. But as all the scores of all relations are required to calculate the loss (one of the correct class and all others to find maximal wrong value) it was decided to take the array of scores as the output and one hot encoded vector as a label. Thus it is both possible to indicate the correct class score and maximal incorrect score in the loss function calculation.

- Due to the form of the loss function, it can start overflow very fast (it uses exponent that is an argument to logarithm and exponent can overflow already with rather small numbers). In the first experiments after first couple thousands of examples all the numbers started to become "nan". In order to fight this problem the work of TensorFlow was converted to "float64" regime, i.e. all the numbers in tensors have this type.

Chapter 4

Experiments

This chapter mainly focuses on answering the following questions:

- What are possible ways of using distantly supervised data?
- Can Distant Supervision improve upon the results of supervised training?
- Can a neural network architecture that is developed in a general domain setting be used to tackle the medical domain - a domain which is usually handled with lots of Natural Language Processing features?
- Which embedding types are best-suited for relation extraction in different domains?

The chapter is divided into two parts - one is dedicated to the supervised learning on manually labeled datasets and serves to demonstrate the baseline, while the second one contains experiments with Distant Supervision. Each of them is divided into a part for the general domain and a part for the medical domain. Overall two supervised datasets for each of the domains were evaluated and one dataset per domain is created for evaluating Distant Supervision. The scheme of all the experiments can be seen in the Figure 4.1.

In order to find the best network configuration, one 4-fold validation experiment was performed once for each domain. The parameters that were tuned were the length of the embeddings, both for words and for distances, and the type of the word embedding. The grid consisted of {30, 40, 50, 70} for distance embeddings length, {300, 400} for word embeddings length and {Swivel, GloVe, Word2Vec} for embedding types. In order to make the model universal, parameters for all the other experiments were fixed to the found via cross-validation.

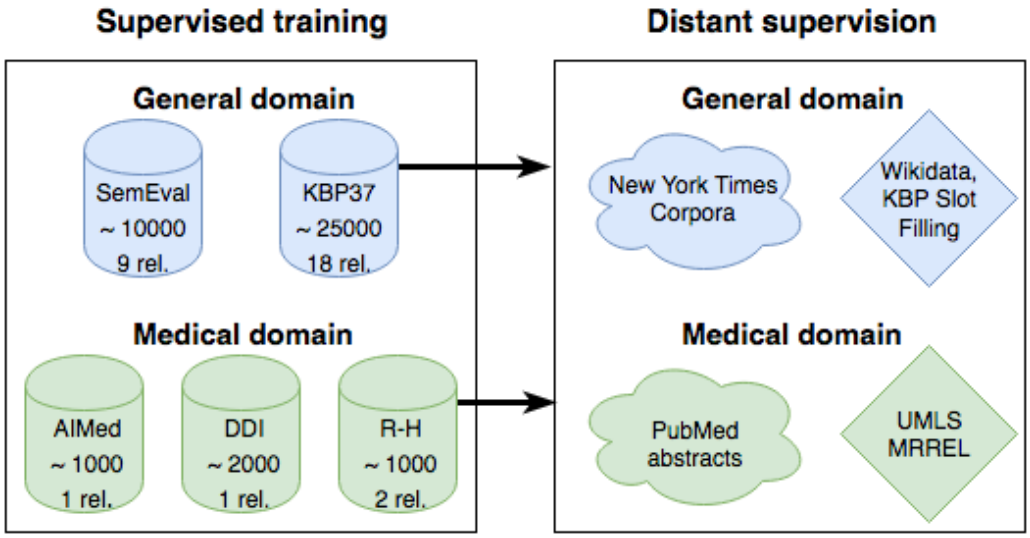


Figure 4.1: Scheme of conducted experiments for the network evaluation.

4.1 Supervised training evaluation, general domain

The goal of experiments in this section is to validate the implementation of the network by comparing with the results achieved in [dos Santos et al., 2015] and prove the quality of the network by applying it to the other supervised dataset in general domain. The experiments in general domain use text corpora and relations from unspecific sources, such as Wikipedia, Freebase and news corpora. The biggest problem in the general domain setting is ambiguity. Named entities can have multiple meanings in different contexts and two entities can also belong to multiple relation types. Generally speaking, detecting one of the general domain relations is a challenging task even for humans. For example, given the entity pair "The Lord of The Ring" and "The Return of the King" it is hard to decide whether the relation between them is "Member-Collection" or "Component-Whole".

4.1.1 SemEval 2010, Task 8

The "SemEval 2010, Task 8" (SemEval) dataset is one of the most common datasets for Relation Extraction evaluation. As the name implies, it was part of the SemEval 2010 competition [Hendrickx et al., 2009], namely for Task 8, and it was also the dataset that [dos Santos et al., 2015] used for

evaluating and training their model. Since the models in this thesis built upon the model from [dos Santos et al., 2015], the SemEval dataset was used to verify the model implementation.

The dataset contains approximately 10000 labeled sentences. The labels are nine relation types and the "Other" class that includes different relations not included in the main ones. The examples were manually collected from the web and annotated in three rounds, so all the annotators would agree on the label given to the sentence. The classes of the relations are following (description is taken from [Hendrickx et al., 2009]):

1. ***Cause-Effect*** *An event or object leads to an effect. Example: those* $< e2 >$*cancers*$< /e2 >$ *were caused by radiation* $< e1 >$*exposures*$< /e1 >$.

2. ***Instrument-Agency*** *An agent uses an instrument. Example:* $< e1 >$*phone*$< /e1 >$ $< e2 >$*operator*$< /e2 >$.

3. ***Product-Producer*** *A producer causes a product to exist. Example: a* $< e2 >$*factory*$< /e2 >$ *manufactures* $< e1 >$*suits*$< /e1 >$.

4. ***Content-Container*** *An object is physically stored in a delineated area of space. Example: a* $< e2 >$*bottle*$< /e2 >$ *full of* $< e1 >$*honey*$< /e1 >$ *was weighed.*

5. ***Entity-Origin*** *An entity is coming or is derived from an origin (e.g., position or material). Example:* $< e1 >$*letters*$< /e1 >$ *from foreign* $< e2 >$*countries*$< /e2 >$.

6. ***Entity-Destination*** *An entity is moving towards a destination. Example: the* $< e1 >$*boy*$< /e1 >$ *went to* $< e2 >$*bed*$< /e2 >$.

7. ***Component-Whole*** *An object is a component of a larger whole. Example: my* $< e2 >$*apartment*$< /e2 >$ *has a large* $< e1 >$*kitchen*$< /e1 >$.

8. ***Member-Collection*** *A member forms a nonfunctional part of a collection. Example: there are many* $< e1 >$*trees*$< /e1 >$ *in the*$< e2 >$*forest*$< /e2 >$.

9. ***Message-Topic*** *A message, written or spoken, is about a topic. Example: the* $< e1 >$*lecture*$< /e1 >$ *was about* $< e2 >$*semantics*$< /e2 >$.

It should be noted, that since sentences in the dataset can contain entities in the different order, there are finally nineteen classes of relations - two for each of the nine main and "Other". Some of the relations are closely connected in order to test the ability to make fine-grained distinctions, e.g. Content-Container, Component-Whole, Member-Collection.

For evaluation of the result achieved by a solver, there is a specially written evaluator, that calculates different characteristics of the predictions made. According to the description in the code of the scorer:

> *The scorer calculates and outputs the following statistics:*
>
> 1. *confusion matrix, which shows*
>   - *the sums for each row/column: -SUM-*
>   - *the number of skipped examples: skip*
>   - *the number of examples with correct relation, but wrong directionality: xDIRx*
>   - *the number of examples in the answer key file: ACTUAL (= -SUM- + skip + xDIRx)*
> 2. *accuracy and coverage*
> 3. *precision (P), recall (R), and F1-score for each relation*
> 4. *micro-averaged P, R, F1, where the calculations ignore the Other category*
> 5. *macro-averaged P, R, F1, where the calculations ignore the Other category*
>
> *Note that in scores (4) and (5), skipped examples are equivalent to those classified as Other. So are examples classified as relations that do not exist in the key file (which is probably not optimal).*
>
> *The scoring is done three times:*
>
> 1. *as a (2*9+1)-way classification*
> 2. *as a (9+1)-way classification, with directionality ignored*
> 3. *as a (9+1)-way classification, with directionality taken into account*
>
> *The official score is the macro-averaged F1-score for (3).*

(taken from the commentaries in the scorer code)

4.1.2 KBP37

The KBP37 dataset [1], as it was called in the paper [Zhang and Wang, 2015], was used to check up the results of solving the problem in the general domain with a fully supervised approach. In contrast to the original paper, the model was evaluated on the SemEval as well as on the KBP37 dataset, to test its perform on other types of relations and entities and to confirm the applicability of the network structure onto different datasets.

The KBP37 dataset is a revision of MIML-RE annotation dataset from [Angeli et al., 2014], that was built from a subset of Wikipedia articles by manual annotation. It contains approximately 25000 labeled sentences. The following changes were made by the authors of [Zhang and Wang, 2015] to adapt it to the description of the SemEval task 8:

- Added direction to the relations, i.e. 'per:employee-of(e1,e2)' and 'per:employee-of(e2,e1)' instead of simply 'per:employee-of'. This is done for all the relations except for 'no-relation'
- Balance the dataset, to exclude the relations that have less than 100 examples for each of the directions. Also, 80% of 'no-relation' examples are discarded
- After that, examples are shuffled and split into three parts, 70% for training, 10% for development and the rest for testing.

After all the modifications, the dataset consists of 18 directional relations, that will result in 37 classes for recognition (two directions for each of 18 and 'no-relation'). This dataset is more complex than SemEval. It has longer sentences (almost twice longer than the longest in SemEval) and also it has multi-relational pairs that are of course common in the real world, but still hard to tackle. Also, it can be observed that both relations and entities in this dataset are more specific. So most of the entities are company or people names, compared to SemEval task, where entities were mostly general objects and people categories (e.g. boy, witch). Furthermore, the relations are very specific, e.g. there are three different classes for placement of headquarters of a company dependent on what it is - a city, a state or a country.

One more aspect of the dataset should be mentioned. Initially, it was labeled by crowdsourcing and the labels were given with different confidence level. But still, the authors of [Zhang and Wang, 2015] took all of the sentences for the training and evaluation. Thus there could be found very imprecise examples, such as:

[1] `https://github.com/zhangdongxu/kbp37`

	Word embedding size 300			Word embedding size 400		
Dist. emb.	GloVe	Word2Vec	Swivel	GloVe	Word2Vec	Swivel
30	78.83	**81.96**	72.20	77.23	60.48	*80.17*
40	78.45	*81.59*	72.03	76.62	61.16	*80.16*
50	77.98	*81.65*	71.50	76.46	62.39	*80.39*
70	77.68	*81.52*	71.10	76.50	62.72	*79.95*

Table 4.1: F1-scores of the cross-validation experiments on SemEval dataset obtained with the official scorer of the competition. Every score is an averaged score of four experiments. With *italics* emphasised largest score for each setup and with **bold** largest in the whole validation.

It was because of $<e1>$*Abu Talib*$</e1>$ *'s (a.s.) good fortune that apart from* $<e2>$*his*$</e2>$ *ancestral services and prestige he also inherited from sons of Ismail (a.s.) high status and courage.* ***per:alternate-names(e2,e1)***

During applying the network to KBP37 'no-relation' class was renamed to 'Other' for consistency with SemEval. The evaluation was performed by adapting the script from the SemEval 2010 Task 8.

4.1.3 Results

Cross-validation experiments were held on SemEval dataset. All parameters, that are not specified as changing for validation testing, were set to the values that are recommended in [dos Santos et al., 2015]. So, an embedding for class "Other" was not trained, the number of filters was set to 1000, the size of the convolution window was set to 3, regularisation rate was 0.001 and learning rate was 0.25, decreasing by dividing by the number of the epoch. Each experiment lasted 15 epochs. The F1-scores of validation experiments can be seen in the Table 4.1. According to the experiment, the best configuration used for all the other experiments in the general domain is a combination of Word2Vec embeddings of the length 300 with distance embeddings of the length 30.

All the train accuracies reached 99.99 - 100 percents, except for Word2Vec of length 400, that gave 96-98 percents. Interesting to note, that independently of the size for the distances embeddings Word2Vec is constantly better than any other embedding type for length 300, but with length 400 Swivel is a stable winner. Also can be noticed, that Swivel embeddings show better results with higher dimensionality, while results with GloVe on the other hand drop. But it should be remembered, that a pre-trained version for GloVe of the length 300 was used. For validating the hypothesis that available pre-

	Word embedding size 300		
Dist. emb.	GloVe	Word2Vec	
30	76.89	62.71	
40	76.48	63.72	
50	76.80	63.31	
70	76.47	63.84	

Table 4.2: F1-scores of the cross-validation experiments on SemEval dataset performed with locally trained versions of word embeddings.

Classifier	SemEval2010	KBP37
CR-CNN [dos Santos et al., 2015]	84.1	-
RNN [Zhang and Wang, 2015]	79.6	58.8
Supervised Ranking CNN	**84.39**	**61.26**

Table 4.3: F1-scores for testing datasets.

trained embeddings are of a higher quality than locally trained ones, the same experiment with locally trained embeddings was held. The results can be seen in the Table 4.2. These results show slightly lower results for GloVe of the length 300, but much lower for Word2Vec.

The best configuration obtained in [dos Santos et al., 2015] was combination of Word2Vec of the length 400 with distance embeddings of the length 70. With this setup they got the best result of $F1 = 84.1$. But conducted cross-validation experiment showed that F1-scores for Word2Vec of the length 400 are worse than all other configurations. The training of these embeddings was done according to the scheme proposed in the paper [dos Santos et al., 2015], but still, score for this configuration is not the highest. Even though training scores for these experiments might give an idea, that training can be continued in order to get better results, further training does not change the score anymore. The score just fluctuates around certain achieved values, going lower and falling back, but not going higher.

The best configuration according to the validation experiments was tested on the test set of SemEval2010 Task8 dataset and KBP37 test dataset. The score achieved is compared to other scores in the Table 4.3. It can be concluded, that model works approximately with same results as in the original paper, so the implementation is correct. Also, it was nice to notice that results achieved with Convolutional Neural Network are higher than with Recurrent Neural Network from [Zhang and Wang, 2015].

Thus, the experiment proved the correctness of the implementation. Also,

the result of the evaluation proves that the model does not perform well only on the one dataset it was constructed for, but also on other datasets as well.

4.1.4 Interpretation

In order to make the work of the network more transparent and explainable, several additional experiments were made.

Representative trigrams As it is described in Subsection 3.3.1 representative trigrams were extracted for every of 18 classes in SemEval2010 and 36 classes in KBP37.

The five representative trigrams with the highest values for each of the classes of SemEval2010 dataset can be seen in the Table 4.4. Should be noticed, that sometimes trigram can be in the beginning or in the end of a sentence or include some punctuation signs. In such cases trigram will not include exactly three words but two or even only one. Also, for one direction of "Entity-Destination" relation there is only one trigram, because the training dataset contained only one example labeled with this class.

Certain conclusions can be made from these lists of trigrams extracted for the relations. First, most of them make sense from the point of view of a human reader. So if a human will try to classify relation "cause-effect" most probably exactly phrases, like *resulted in* or *caused by*, would be a sign that a sentence contains this relation. Second, some of the trigrams still contain somehow specific words, for example, "bottle", "suitcase", "box" for "content-container" relation. In order to understand the origination of such words, the most frequent entities in the dataset were looked up. And it revealed, that "suitcase" for example is a very frequent entity for "content-container" examples, with 21 out of 433 examples for one direction and 31 out of 181 for the other. Thus, if some word appears very frequently in all examples for the relation it most probably will be among representative trigrams.

Most of the trigrams extracted for KBP37 dataset are not very general. For example, for class cities-of-residence the most valued trigrams are "in Los Angeles, in Detroit Michigan, in London in, to London to", so basically universal parts are only "in" and "to", everything else is very specific to the dataset. One can assume that this happens because of the very close meanings and entity pairs in all examples for one particular class. At the same time, for example, class "founded-by" is quite generalisable through its representative trigrams, such as "founder of the, created google in, founder of gome, founders of dow". Another explanation that entities in this dataset are mostly proper names, so in order to recognise relation it would be easier

Relation	(e1, e2)	(e2, e1)
Message-Topic	the news that, contains a description, laws defining, of publications discussing	topic of conversation, topic of discussion, been reflected in, been discussed in
Cause-Effect	common cause of, that resulted in, main causes of, leading causes of	are caused by, been caused by, was caused by, is caused by, damage caused by
Component-Whole	part of the, the crank of, lid of the, handgrip of the, part of this	the knife blade, has a coil, my ear lobes, the mouse button
Entity-Origin	was distilled from, is derived from, popped out of, is distilled from, away from the	the source of, of wheat liquor, some strawberry syrup, on rye liquor
Member-Collection	essays collected in, in the army, the head of, of the team, a soldier joins	a confederacy of, a cooperative of, a federation of, a cabal of, covey of partridges
Instrument-Agency	a crane operator, the elevator operator, are used by , a forklift operator	with a spoon, the author uses, a person applies, potter 's wheel
Product-Producer	produced by the, book 's author, created by the, from the author, founded by a	the factory's, issued a statement, factory's output, the designer's
Entity-Destination	poured flour into, was put into, were released into, poured water into, are migrating into	and steadily climbs
Content-Container	in a box, in a suitcase, in a crate, in a bottle, money was in	a bottle full, a bottle with, a suitcase full, a suitcase with, envelope contained a

Table 4.4: Representative trigrams according to the network for SemEval2010 Task8 dataset classes.

to learn the own names for example. Moreover, a look into the dataset reveals that sentences for one and the same relation really do not have a lot of "characteristic" words in common, like it was in the case of SemEval dataset. It is also noticeable, that overall Precision is higher than Recall, i.e. network learned some specific details, that allow distinguishing some examples nicely, but not to see all possible examples of the relation.
org:founded-by *founder huang guangyu, the congregation of, founder bill gates, international pictures co-founder, founder of the, schlafly 's eagle, created google in, founder of gome, founders of dow*
per:alternate-names *born luigi curto, as mr. x., formerly baldwin ii, bob dunn was, name lal is, name cassidy is, name dresta is, force his ouster, that his relationship*
org:members *(nyse :, cent lloyd banks, coached ncaa men, 's division i, ncaa division i, the university of, big east conference, 2011 ncaa division, 2012 ncaa division*
org:top-members/employees *chief executive officer, leader stockwell day, british vogue editor, victoria police chief, chief executive of, the army of, chief executive officer, alexandra shulman editor*
per:countries-of-residence *united states ., in france ., of canada ., prime minister of, nepal (maoist, greece karolos papoulias, australia 's first, botswana president ian, nicaragua daniel ortega*
org:founded *in 1989, in 1998, october 2005, in 1997, in 1956 and, in 2003 by, in 2005 with, in 1982 as, in 1998 the, in 1998 and*

As direction of relations are not clearly distinguishable in representative trigrams, they were united together. Not all the relations were included, as the general idea already clear. Here the same check for the most frequent entities revealed again that most of them will be in the trigrams. So for "org:members" "division_i" is the most frequent entity with 10 out of 426 examples and also "ncaa" with 54 out of 736 examples for the other direction.

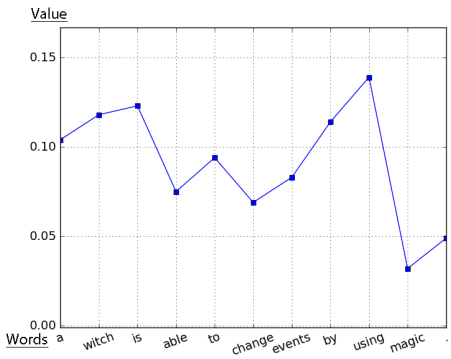
Overall it might be concluded, that for improving the quality of the training some kind of entity replacement might be applied. I.e. all the entities should be replaced with one and same word (Entity1 and Entity2 for example), so the network will not be able just to memorise entity names in order to recognise the relation.

Semantic values The second experiment is done according to Subsection 3.3.2. It makes conclusions about the semantic values of the words in the sentence for the answer given by the network. For the experiment three prototypical sentences were randomly chosen from the test set of SemEval2010

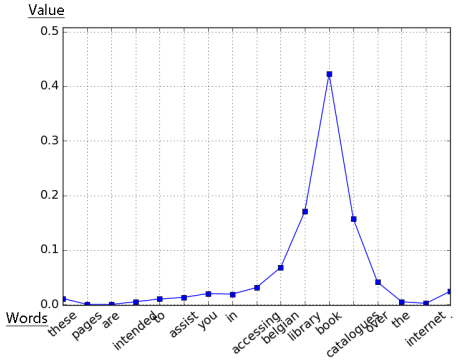
dataset, the first one was classified correctly, the second one was classified alternatively (i.e. both labels are considered to be correct according to the dataset authors) and the last one was classified wrong. They are:

- "A $<$ $e1$ $>$witch$<$ $/e1$ $>$ is able to change events by using $<$ $e2$ $>$magic$<$ $/e2$ $>$."
- "These pages are intended to assist you in accessing Belgian library $<$ $e1$ $>$book$<$ $/e1$ $>$ $<$ $e2$ $>$catalogues$<$ $/e2$ $>$ over the internet."
- "The $<$ $e1$ $>$plant$<$ $/e1$ $>$ grows from an underground $<$ $e2$ $>$storage unit$<$ $/e2$ $>$ called a corm."

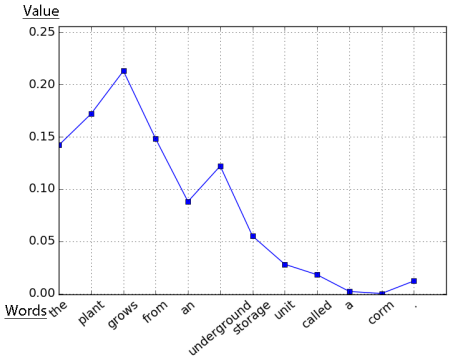
Corresponding plots for these examples can be seen in the Figure 4.2. The first sentence recognised correctly as Instrument-Agency with a spike on "by using". So it means that the network correctly learned the connection between syntactic structure "by using" and relation Instrument-Agency. The second sentence was classified as Message-Topic, while according to the label it is Component-Whole. But the comment to this example in the original dataset says that both of the classes can be considered to be correct. Here no characteristic words are present, so only the entities give spikes in the plot. The third sentence was recognised incorrectly as Other, while the correct label is Entity-Origin. It is seen that spikes are on words "grows from" but still the values were not enough to gain the maximal score for the correct class. It can be explained if one checks the frequency of occurrences of the "spiked" words. So *from* is very frequent for several relations, such as "Cause-Effect", "Product-Producer", "Other" and the correct relation "Entity-Origin" as well. But *plant* is more frequent for "Product-Producer" and *grows* for "Other". So finally network was confused.



(a) witch - magic



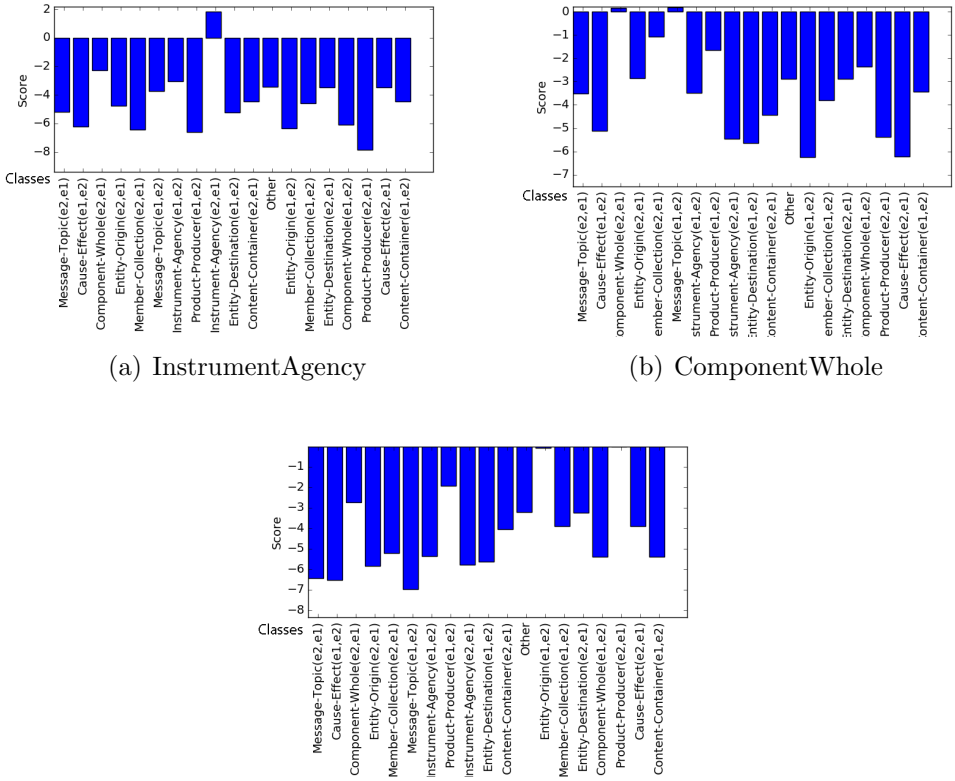
(b) book - catalogues



(c) plant - storage unit

Figure 4.2: Corresponding plots for three selected sentences from the test set of SemEval2010 for calculation of semantic values for words in the sentence.

Scores distribution This experiment is described in Subsection 3.3.3. It helps to look into the differences of scores given by the network to the relational classes. The Figure 4.3 shows it for the same three examples, that were used for the experiment with semantic values. The first sentence is correctly classified and it can be seen, that all other scores are negative and only the correct class gives a comparatively large positive score. The second sentence distribution shows that network was hesitating between two acceptable classes and thus it got two small, but positive scores for both of them. The third sentence was recognised as "Other", thus all the scores have to be negative. But two classes got very small negative scores, that are correct class "Entity-Origin" and very close to the context of the sentence "Product-Producer".



(a) InstrumentAgency

(b) ComponentWhole

(c) EntityOrigin

Figure 4.3: Scores distributions of relational classes for three chosen sentences from the test set of SemEval2010.

4.2 Supervised training evaluation, medical domain

These experiments were held to check the applicability of the ranking CNN to the medical domain. Thus several popular medical datasets for Relation Extraction were chosen and evaluation compared to existing results. There are a lot of relational classes specific only for medical domain. One of the most popular ones is about a general interaction between different entities, such as proteins, genes, drugs. Each interaction can have numerous more specific subclass relations, but for now tests are performed on generic relations. One more considered dataset contains *treatment-for* and *prevents-from* relations, that are very interesting by having high granularity level, i.e. they are very close by meaning.

4.2.1 AIMed

AIMed dataset was a result of research in [Bunescu et al., 2005]. The authors are concerned with automated information extraction from medical texts, especially information about human genes/proteins. They chose to think of genes and proteins as interchangeable ideas because there is a direct correspondence between them. To obtain the dataset, the authors manually tagged about one thousand Medline abstracts [2]. They mention following sources of abstracts for human protein interactions:

1. 200 abstracts that are known to contain protein interaction from the Database of Interacting Proteins

2. 30 abstracts for negative examples, where more than one protein mentioned, but there is no interaction between them

In such a way, the AIMed dataset consists of abstracts with marked proteins and protein interactions as pairs. For example

```
In contrast , in the absence of <prot>  p21ras </prot>
, coexpression of <prot>  JAK2 </prot>  and <prot>
Raf - 1 </prot>  resulted in an overall decrease in
the <prot>  Raf - 1 </prot>  kinase activity .
```

has several marked proteins, but no relations. And this

```
Under these conditions , a ternary complex of
<p1  pair=6 >  <prot>  p21ras </prot>  </p1>  ,
<p2  pair=6 >  <prot>  JAK2 </prot>  </p2>  ,
and  <prot>  Raf - 1 </prot>  was observed .
```

has three proteins and one interaction pair in it. Since the experiments require sentences with only two marked entities the dataset was preprocessed before usage. Each marked entity in the sentence was considered as the first entity with each other entity as the second one. Each pair was considered only once in the order they are mentioned. An example was marked positive if proteins were in the same "pair" number as "p1" and "p2" nodes and negative otherwise. In the case of presence of "p1" or "p2" tags, only proteins in these tags were considered, as the number of negative samples is significantly larger

[2] https://www.nlm.nih.gov/bsd/pmresources.html

anyway. So from the first aforementioned sentence six negative examples would be obtained, with entity pairs "p21ras" - "JAK2", "p21ras" - "Raf - 1" (first occurrence), "p21ras" - "Raf - 1" (second occurrence), "JAK2" - "Raf - 1" (first occurrence), "JAK2" - "Raf - 1" (second occurrence), "Raf - 1" (first occurrence) - "Raf - 1" (second occurrence). In the case of the second sentence, one positive example would be generated with entity pair "p21ras" - "JAK2".

The dataset is highly unbalanced, having much more negative examples, than positive ones. But as all the evaluation tests in other papers are done without balancing, so were held the experiments in here.

4.2.2 DDI

The second important relation in medical domain is a drug interaction. The drug interaction is observed when one drug influences the level of activity of the other drug [Segura Bedmar et al., 2011]. Knowledge about drug reactions is critical for patient safety and healthcare costs. There are drug databases, for example, DrugBank database [3], but data in them is not always up to date. New information is published regularly in reports and articles. The authors of [Segura Bedmar et al., 2011] are concerned with the application of automatic Natural Language Processing tools for extracting knowledge about drug interactions from textual data. Therefore gold standard dataset for training machine learning models was created. This dataset consists of manually annotated with interactions DrugBank database texts, where drugs were marked by MetaMap tool [4]. The example of an annotated sentence is:

```
<sentence id="DrugDDI.d27.s0" origId="s0" text="The concomitant
  intake of alcohol and Acamprosate does not affect the
  pharmacokinetics of either alcohol or acamprosate.">
<entity id="DrugDDI.d27.s0.e0" origId="s0.p1" charOffset="26-33"
 type="drug" text="alcohol"/>
...
<pair id="DrugDDI.d27.s0.p0" e1="DrugDDI.d27.s0.e0"
e2="DrugDDI.d27.s0.e1" interaction="false"/>
...
</sentence>
```

So for each sentence drugs are marked and then all possible pairs with indication of interaction between entities - either "true" or "false". For the test

[3] https://www.drugbank.ca/

[4] https://metamap.nlm.nih.gov/

dataset, interaction is set to "?" and in separate file with gold annotations the labels are written down.

In order to use the dataset each sentence was repeated as many times as it has pairs and for each case only one pair of entities was marked. Labels were given according to the corresponding interaction label. Again, as with AIMed, the dataset is very unbalanced, having more negative examples than positive. But as evaluation papers use exact unbalanced test set balancing was not performed.

4.2.3 Rosario-Hearst dataset

This medical dataset was developed for the research in [Rosario and Hearst, 2004]. Initially, the dataset was obtained from MEDLINE abstracts and manually labeled by an expert for seven possible relations between DISEASE and TREATMENT. Some of the sentences among the examples were also labeled as "only disease" or "only treatment" and some identified relations, such as "no cure" had too few examples. Thus the dataset was adapted for making experiments with Ranking CNN. First, all not binary relations were filtered out. Then among them only large enough were left, they are "treatment for" and "prevents from". All the other binary relations were united for the class "Other". Finally, the dataset includes 810 sentences labeled as "treatment for", 63 sentences labeled as "prevents from" and 69 for "Other". Dataset also contained "not relevant" examples, i.e. sentences that simply do not contain any entities or relations at all. These examples were outnumbering any of classes, but they could not be used for the experiments with the Ranking Convolutional Neural Network.

This dataset was initially represented as a set of sentences with two (or less) marked entities for each sentence, so the transformation for the experiments was straightforward.

4.2.4 Results

Validation tests were held on AIMed dataset at the same setup as for general domain. But also one more choice was tested - usage of general domain embeddings and embeddings trained on PubMed abstracts dump from December 2016. The resulting F1 scores could be seen in the Table 4.5.

All the training accuracies reached 100 percent, sometimes after 4-5 epochs, except for experiments with Word2Vec of length 400 trained on general domain. In those experiments accuracy was reaching 94-96 percents. From the results of the evaluation, it can be seen that embeddings trained on medical domain always give better result. The highest results are also

	General domain embeddings					
	Word embedding size 300			Word embedding size 400		
Dist. emb.	GloVe	Word2Vec	Swivel	GloVe	Word2Vec	Swivel
30	92.295	90.04	92.12	91.86	86.72	92.76
40	92.49	89.74	92.05	92.09	85.49	93.00
50	92.36	89.19	91.87	92.09	85.13	91.27
70	92.62	87.65	92.07	91.797	87.099	91.32

	Medical domain embeddings					
	Word embedding size 300			Word embedding size 400		
Dist. emb.	GloVe	Word2Vec	Swivel	GloVe	Word2Vec	Swivel
30	91.97	91.71	92.43	91.65	91.56	*93.25*
40	92.44	*92.13*	*92.72*	**93.38**	91.23	92.46
50	92.19	91.397	92.51	91.85	91.92	92.55
70	*92.84*	90.95	92.47	92.19	*92.197*	91.7

Table 4.5: Results of the cross-validation experiments on AIMed dataset obtained by simple calculation of F1-score for binary case. Every score is an averaged score of four experiments. First table contains results for general domain embeddings, second for medical domain. With *italics* marked scores maximal in the column and **bold** is for the maximal result overall.

always obtained with GloVe embeddings, while changing the dimensionality of word embeddings does not critically change the score. Change in the size of distance embeddings causes worse results when getting higher than 50. As the best model was chosen the one with GloVe medical embeddings of size 400 and with distance embeddings of 40.

As in the original paper [Bunescu et al., 2005] results were obtained by 10-fold cross-validation and models in several other papers were as well ([Bunescu and Mooney, 2005], [Airola et al., 2008]) evaluated in this way, the test for AIMed dataset was also performed with original 10-fold split. The DDI dataset also contains direct split on training and testing articles, so all the sentences formed from the testing articles were used as a testing set. The Rosario-Hearst dataset was simply split to 75% of training data and 25% of testing data and evaluation was held on the testing data. The results of testing the network are in the Tables 4.6, 4.7, 4.8.

[5]As in both [Bunescu et al., 2005] and [Bunescu and Mooney, 2005] authors gave PR-curve for result evaluation here was used the point corresponding to P=55.69 as obtained by the Ranking CNN.

[6]Training was performed on the dataset where number of negative and positive examples was balanced by cutting off negative examples.

[7]Results of competition from the original paper [Segura Bedmar et al., 2011]

Classifier	P	R	F1
Original paper best result [Bunescu et al., 2005] [5]	55.69	36	43.73
Subsequent kernels [Bunescu and Mooney, 2005]	55.69	54	54.83
Graph kernel [Airola et al., 2008]	52.9	61.8	56.4
Supervised Ranking CNN (balanced train) [6]	34.27	**90.56**	49.29
Supervised Ranking CNN	55.69	63.85	**58.38**

Table 4.6: Results of evaluation of the network with AIMed.

Team [7]	P	R	F1
WBI	60.54	71.92	65.74
FBK-HLT	58.39	70.07	63.70
Uturku	58.04	68.87	62.99
LIMSI-CNRS	55.18	64.90	59.65
laberinto-uhu	50.00	44.37	47.02
Supervised Ranking CNN (balanced train)	22.38	**100.00**	36.58
Supervised Ranking CNN	**91.75**	98.68	**95.09**

Table 4.7: Results of evaluation of the network with DDI.

For the training datasets, all the experiments were getting 100 percent accuracy.

It is interesting to notice, that training on the balanced dataset while testing still on unbalanced leads to higher Recall, but lower Precision. It directly reflects the idea, that distribution of both training and testing sets should be the same, otherwise supervised learner will not be able to be precise. So having a balanced training set, model suggests that much more examples are positive in the testing set than it really is - from what follows high Recall. But this time it is not the case and that is why Precision is low.

Also, it should be noticed, that in the AIMed and DDI datasets one and the same sentence will be an example for both negative and positive labels. And of course, it is hard for the network to extract syntactic construction that will be a sign of positive relation.

So it can be concluded, that the model gives state-of-the-art results for

Classifier	P	R	F1
Supervised Ranking CNN	90.05	82.14	85.58

Table 4.8: Results of evaluation of the network with Rosario-Hearst dataset.

various medical datasets with different classes of relations. Compared to the evaluation from several other authors Ranking CNN gives always a better result.

4.2.5 Interpretation

Representative trigrams extraction In order to see the effect of balancing datasets, several sets of representative trigrams were extracted.

The first one is for the network trained on the balanced AIMed dataset, second one is for the unbalanced dataset and the third one was made as a separate experiment, where dataset was initially balanced and then split on 75% for training and 25% for testing. The result shown by this training was P=51.91, R=92.08, F1=66.39.

Balanced dataset *interaction with p25shc, binding, complex, interaction*

Unbalanced dataset *interaction of the, interaction with p25shc, binding, binding region of*

Initially balanced dataset *tr6 specifically binds, interacts poorly with, interacts with the, interaction*

From the human point of view, all the sets capture a lot of meaningful expressions of interaction.

The same three sets were obtained for DDI dataset. The scores for the network trained on the initially balanced dataset are P=59.68, R=98.84, F1=74.42, so it performs better than on AIMed dataset. But of course, it should be noticed that in general results for DDI dataset are also higher.

Balanced dataset *of aprepitant with, hormonal contraceptives may, of erythromycin and, taking tarceva with, of erythromycin with, probenecid is, amphetamines may, azithromycin had, the benzodiazepines may*

Unbalanced dataset *etonogestrel may interact, monoamine oxidase inhibitors, barbiturates, of erythromycin with, of amphetamines, nsaids can reduce, carbamazepine, may inhibit, nsaids may diminish*

Initially balanced dataset *caution, agents, drugs, inhibitors, amphetamines, levels*

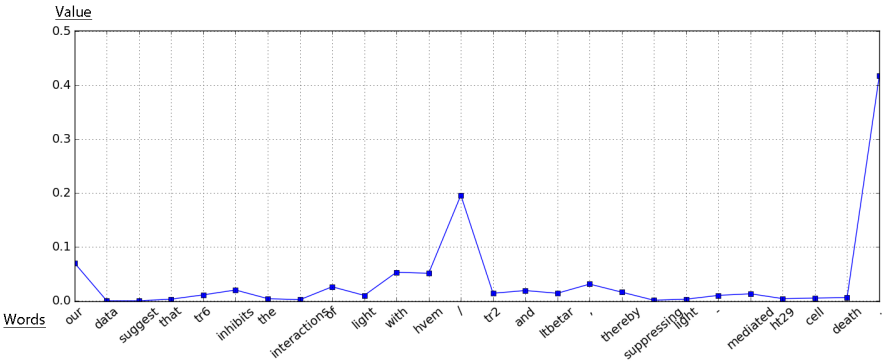
For DDI dataset most meaningful are trigrams extracted for the unbalanced training set. It can be explained by considering the fact, that the unbalance in amounts of negative and positive examples for this dataset are higher (almost 15 times more negative examples), than for the AIMed (around 3 times more negative examples). So when balancing is performed, network learns more drug names that can affect each other, rather than syntactic constructions denoting interactions.

Semantic values For checking how the network "sees" the sentence in terms of values for the final decision two examples from AIMed test dataset were taken. The first one was classified correctly and the second one was mistakenly classified as "Interaction". The sentences are:

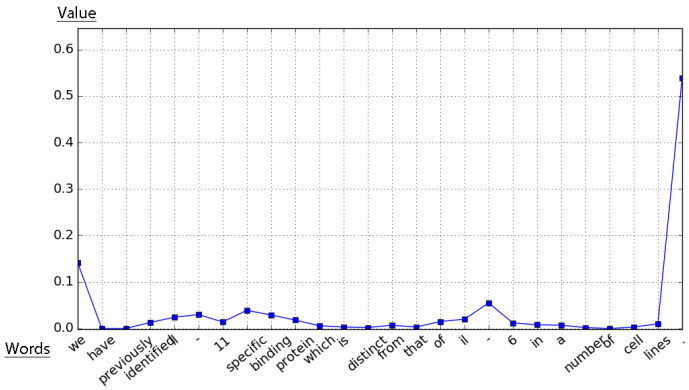
- "Our data suggest that TR6 inhibits the interactions of $<$ $e1$ $>$LIGHT$<$ $/e1$ $>$ with $<$ $e2$ $>$HVEM$<$ $/e2$ $>$ / TR2 and LTbetaR, thereby suppressing LIGHT - mediated HT29 cell death."
- "We have previously identified $<$ $e1$ $>$IL - 11$<$ $/e1$ $>$ specific binding protein which is distinct from that of $<$ $e2$ $>$IL - 6$<$ $/e2$ $>$ in a number of cell lines."

Corresponding plots for these examples can be seen in the Figure 4.4. The first sentence classified correctly as denoting "Interaction" and it can be seen that there are spikes over *inhibits* and *interactions of* and *with*. As the value calculated for the trigram, it would be visualised over the central word of it. Thus, spike in the centre over / is a spike over a protein name *hvem/tr2*. That denotes that network is learning entities names as well. The second example is classified wrongly as describing interaction. The plot has numerous spikes over names of proteins and over *specific binding protein* that leads to the high score for the "Interaction" class, while these exact proteins do not interact.

Additionally, it is seen, that with longer sentences semantic values become much smaller (compare to analogous evaluation for SemEval2010 dataset in general domain) because they are distributed among all the words and thus it is harder to extract needed parts. Due to this spikes over the beginning and the end of an example become more prominent. They appear because sentences are padded with zero values for convolution and the activation function is tanh. Thus, zeroes can get higher values than some other words in the sentence with negative values.



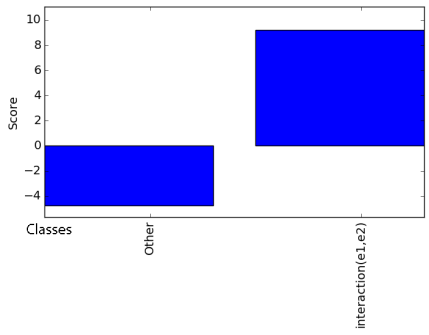
(a) LIGHT - HVEM



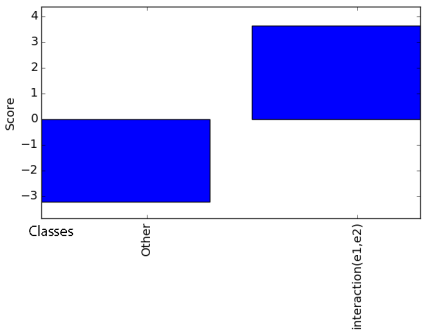
(b) IL-11 - IL-6

Figure 4.4: Semantic values for classification of the words in sentences.

Scores distribution Corresponding scores distribution plots for the same two examples can be seen in the Figure 4.5. The first example was recognised correctly and the network is quite sure in its answer. The second one was recognised incorrectly with a rather high certainty as well. This again can be explained by the spikes on the protein names that are seen on the semantic evaluation plot for the second example.



(a) Interaction



(b) Other

Figure 4.5: Distribution of scores values.

4.3 Distant supervision evaluation

The question to answer by these experiments is the possible quality of the model trained distantly. Also, possible ways of creating distantly supervised data are described.

4.3.1 General domain: KBP37 based distantly supervised dataset

As it was mentioned before KBP37 dataset was adapted from the MIML-RE dataset from [Angeli et al., 2014]. In order to evaluate Distant Supervision in general domain, this dataset was chosen, as authors of [Angeli et al., 2014] made publicly available the set of relation pairs that were used for the creation of the dataset. They used 2010 and 2013 TAC KBP [8] documents as a source of relations. For creating the supervised dataset they utilised Amazon Mechanical Turk [9] for labelling examples that they got after aligning entity pairs to sentences from 2013 Wikipedia dump. They annotated 23725 examples, that were transformed to KBP37 dataset by [Zhang and Wang, 2015].

A closer investigation of the entity pairs from TAC KBP showed that the quality of them is not high enough. For example, they could contain only one letter as a name or just a word "president" as an entity for "org:top-members/employees" relation. Thus as a Knowledge Base for Distant Super-

[8] `https://www.ldc.upenn.edu/collaborations/current-projects/tac-kbp`
[9] `https://www.mturk.com/mturk/welcome`

vision experiments both relational pairs from MIML-RE [10], i.e. from TAC KBP, and Wikidata [11] were used. Wikidata returned fewer pairs corresponding to required relations, but they are more precise. Entity pairs from Wikidata were queried through the query interface for each of the corresponding relations. The amount of entity pairs for each of the relation types varies a lot - from less than 1000 to more than 50000. For creating the "Other" class were chosen "per:religion", "per:children", "org:political/religious-affiliation" entity pairs. In order to minimise noise effect of not accurate entity pairs from TAC KBP, the entity pairs containing one letter entities or names consisting only of capital letters with dots were filtered out and not used.

For raw text corpus the NewYork Times archive [12] was taken. According to the idea of the Distant Supervision an alignment of entity pairs from a Knowledge Base to the text should be performed. This was done by the simple method of string matching. It means that entities' names were simply searched in the textual corpus. If sentence included two entities that are related according to the Knowledge Base that would give an example for the corresponding relation. In order to implement the lookup of the words in a large volume of textual data, Whoosh, a python library for indexing the text, was used [13]. It allows to index each sentence of the text as a separate entry and then make a request to search for two entities' names and returns all the entries containing them. So, raw textual data from NYT corpora was split into sentences with NLTK library, indexed by Whoosh and then aligned with the set of entity pairs from the Knowledge Base. If all of the aligned sentences are taken, it would lead to a highly unbalanced, both by the amount of examples per relational class and the amount of examples per entity pair, and really huge dataset. That is why only 5 examples per entity pair were taken and the maximal amount of sentences in relational class was set to 3000. This resulted in a dataset with the overall amount of 75913 sentences. And even this dataset is already almost tree times more than the supervised dataset for the same relational classes and it was obtained completely automatically. It proves the point, that the amount of distantly supervised training data can be made as large as needed without any additional effort.

Apart from simple training on the distantly supervised data, several experiments for quality improvement were held:

- Applying Multiple Instance Learning. As it was explained in Subsection 3.2.1, training examples for this setup are bags containing at least

[10] `https://nlp.stanford.edu/software/mimlre.shtml`
[11] https://query.wikidata.org/
[12] `https://catalog.ldc.upenn.edu/ldc2008t19`
[13] `https://whoosh.readthedocs.io/en/latest/`

Experiment	P	R	F1
Supervised training	**67.74**	57.88	**61.26**
Distantly supervised training	50.71	45.24	43.81
Distantly supervised + MIL	51.82	46.61	45.40
Distantly supervised + supervised data	57.64	57.84	55.03
Distantly supervised + supervised data + MIL	60.25	**58.24**	56.93
Transfer from supervised	48.93	44.14	42.73
Transfer from supervised + MIL	51.13	44.92	44.58

Table 4.9: Precision, Recall and F1-scores for distantly supervised training evaluation.

one correctly labeled sentence. Thus sentences were grouped based on the entity pair they contain and each of this bags was labeled as a relation, that this entity pair has in the Knowledge Base.

- Mixing supervised data. It was mentioned in [Riedel et al., 2010] that distantly supervised data introduces noise, but it was decided to check how adding existing supervised data into the distant dataset would affect the training results.

- Using transfer learning. Generally, it might be a good idea to train the model on existing supervised data and then continue tuning it with more and more distantly supervised data to get better results.

4.3.2 Results

The results of training the network in various ways with distantly supervised data can be seen in the Table 4.9. For comparison, the supervised training result was also included in the table.

The first conclusion that can be made - training without manually labeled data can be performed and it will achieve definitely higher results than random assignment (with the amount of classes 37 F1-score for random assignment would be around 0.2%). Thus it was successfully proven, that having publicly available Knowledge Base with required relation types and a large amount of textual data training set for a neural network solving the task of Relation Extraction can be constructed automatically. In the context of the task to continuously extract new knowledge from newly published texts, this approach is more appealing than manual extraction. Supervised training might give better results but it still requires the manually curated creation of training dataset, that is most of the times not possible and also should be

repeated for any new relational class. Also, in real-world application, while validating the results for including into a Knowledge Base not only the top score answer might be considered. Thus, if only top one considered as an answer ratio of correct answers is 42.99%. If top three answers are considered for checking, then already 68.14% will be recognised correctly. Further addition does not change the result so much - with top five the percentage of correct answers is 79.58%.

The second point is a definite positive effect of Multiple Instance Learning. Every time when Multiple Instance Learning was applied the previous result was improving by approximately two percents. Good point also that not only Precision or Recall alone is improved, but both simultaneously. It can be concluded, that exploration of further more complex methods for noise reduction in the distantly supervised data will significantly improve the score.

The results of transfer learning experiment were disappointing, it performs even worse than simple distantly supervised training. It might be explained by different directions that datasets are trying to lead the network - scopes of syntactic constructions in supervised dataset and in distantly supervised dataset might differ a lot. Thus pre-training with supervised data leads to a worse starting point for distantly supervised training than a random one.

As it was expected, mixing in existing supervised data improved drastically distantly supervised training. And also as expected, it gives a worse result than pure supervised training. But it should be pointed out that with Multiple Instance Learning together with mixed dataset setup, it was possible to reach higher Recall than for supervised training. This means that distantly supervised data helps the model to become more general, be less biased by the provided supervised dataset. It again might be a very good sign for live models, that are supposed to retrieve more and more new knowledge from new textual data, because it is never possible to provide the needed amount of new labeled training data manually.

4.3.3 Interpretation

Distantly and manually supervised datasets It is important, that manually supervised training and testing datasets are tightly coupled and they will have common context and common biases. Thus, evaluating distantly supervised model with existing testing dataset might be not objective. There exist other ways to evaluate the results of Distant Supervision, for example, performed in [Mintz et al., 2009], but they would not show a realistic comparison to the supervised results. Moreover, evaluation with supervised testing dataset allows having a fresh look at the quality of the supervised

data. It might not be absolutely accurate, especially when the number of examples is large. And the worst aspect is that the supervised training will cause bias to specific kind of mistakes (as they will be both in training and testing set).

Overall, if there exist a hypothetical full set of all possible syntactic constructions for describing a specific relation, supervised dataset would be some (most of the times not very big) subset of it and of course, it might contain errors of labelling, that both training and testing sets would share. On the other hand, distantly supervised dataset would be slightly different subset of the whole set, that might be spread more and more - thus evaluation with initial supervised testing set would be less and less objective, because some constructions that are not there would be recognised by the price of forgetting constructions from the supervised dataset. In general, this idea is reflected in the Figure 4.6.

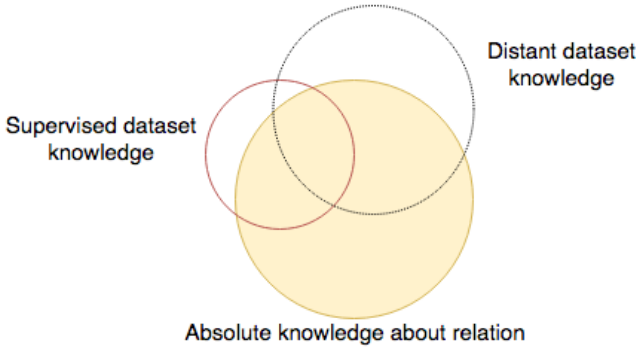


Figure 4.6: Concept of knowledge about a specific relation contained in supervised and distantly supervised datasets.

It seemed quite logical to suggest, that in order to have more basement for such conclusions one should look into the ratio of correctly recognised examples for manually supervised and distantly supervised networks. The result of this investigation is depicted in the Figure 4.7. It can be seen, that for some of the relations manually supervised network will recognise much more constructions, but there are still relations for which distantly supervised network finds more constructions. Overall, a number of examples recognised only by the distantly supervised network is quite sensible. It can be concluded that the information contained in the distantly supervised dataset is useful and quite different from the one in the manually labeled dataset. Thus, the main goal when working with distantly supervised dataset is to get rid of the noise that lowers the precision of recognition.

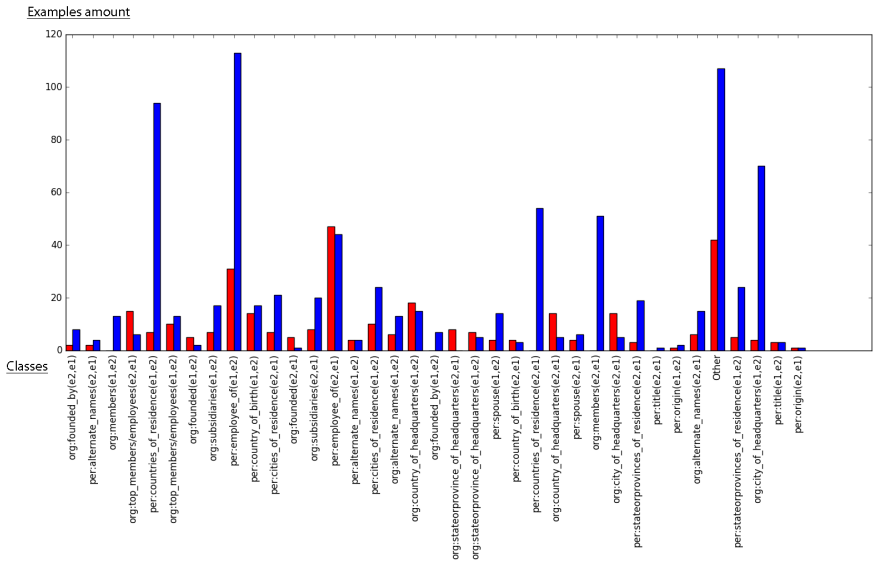
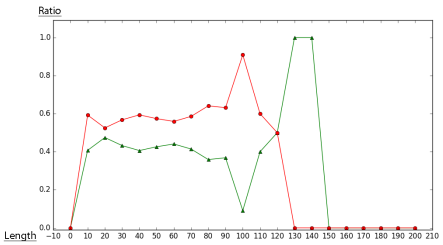


Figure 4.7: Blue bars show amount of testing examples that were recognised correctly by manually supervised network, but were not recognised by distantly supervised one. Red bars show the same amount for distantly supervised network.

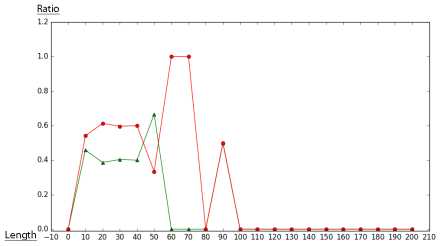
Length of the examples Of course one more very important aspect of relation recognition in a sentence is the length of the sentence and the distance between the entities in it. The dependency can be seen in the Figure 4.8. Spikes around the large values of length and distance are not representative, as the number of the examples there much smaller (3-5 sentences). But the overall the tendency is clearly seen - with the enlargement of the length or distance a number of errors grows and a number of right answers drops. Any distantly supervised dataset will always be characterised by longer sentences on average, so this aspect should be taken into account when the dataset is constructed. For example, sentences longer than some limit can be simply not included in the final set of training examples.

Representative trigrams If one takes a look at representative trigrams for the model trained on the distant dataset, mixed with supervised data and united in bags, several interesting observations can be made. For example for relation "per:spouse", while supervised dataset obtained:

king george iii, his second wife, married gene raymond, wife melanie griffith, wife of actor, wife henrietta maria, husband jeff richmond, wife kimberly



(a) Sentence length dependency



(b) Distance between entities dependency

Figure 4.8: Dependency of the number of correct answers (green) and wrong answers (red) normalised by the overall amount of examples of specific length or with specific distance between entities correspondingly on the input sentence length and on the distance between entities.

williams-paisley, wife of president

newly trained model was able to obtain more general trigrams, such as:

wife of the, husband the, wife the, wife lee radziwill, wife of president, married yesterday to, wife of senator, wife of prime

And the similar situation can be observed for several other relations. For example, "per:cities-of-residence" becomes less unbiased in a sense of taking into account more cities, than it is in the training set.

Active learning Even more interesting is an idea of exploiting representative trigrams as a tool for improving dataset. Thus, a look into representative trigrams obtained for the distantly supervised network can reveal some learned aspects that do not make sense from the human point of view. For example, for the relation "org:founded-by" found trigrams *open society institute, fox broadcasting company, ethical treatment of, jack daniel's* seem to be not really representative. Same for the relation "per:alternate-names" a trigram *mimi smith*. If these non representative trigrams are found in a sentence from the training dataset, the sentence might be filtered out, so the network will not learn the trigram anymore. A small experiment with two aforementioned relations showed improvement in scores. So, for "org:founded-by" Precision grew from 54.05 to 70 and Recall from 25 to 26.25. For "per:alternate-names" Precision improved from 33.33 to 38.24 and Recall from 21.74 to 28.26. This is a possibility to further improve Distant Supervision by application of Active Learning and iterative transformations of the training dataset.

4.3.4 Medical domain: Rosario-Hearst based distantly supervised dataset

The Distant Supervision experiment setup for the medical domain was mainly performed according to [Roller and Stevenson, 2014]. Rosario-Hearst dataset was chosen as base supervised dataset because it contains more than two relations and relatively small that can allow evaluating the effect of adding distantly supervised data more.

As a textual corpus, PubMed abstracts were used. The abstracts were downloaded without specialised search and only around 500000 were finally used for creating the dataset. In order to mark medical entities, MetaMap online tool [14] was utilised. It allows to setup a lot of parameters for different output formats and text processing. As a simplest one, the Fielded MMI format was chosen. This format gives as an output all found entities, their unique identifiers in the MetaMap system Knowledge Base, their positioning in the text and additional information that was not used. It was chosen as it gives the least volume of the processed file and rather simple to use further. Also, it is important to notice that the input file was preprocessed with sentence tokeniser from NLTK toolkit, thus output was directly giving positions of the entities in each of the sentences.

The next step is to align a publicly available Knowledge Base to the preprocessed text. As entities marked with identifiers from MetaMap, relational base MRREL from the same source was used. This Knowledge Base [15] consists of many datasets and thus includes a lot of relational classes (almost 600). Among them "may-be-treated-by" and "may-be-prevented-by" which correspond to "treatment-for" and "prevents-from" relations from the supervised dataset. So, entity pairs are taken from these two classes of MRREL Knowledge Base. Then, with a Python script, all the tagged sentences that contain entity pairs with desired relations are separated. In order to construct examples for "Other" class, all possible pairs of entities that are not in one of the classified relations according to the Knowledge Base were considered. Thus, a dataset containing 13025 sentences was formed. For medical domain difference in sizes of the manually and distantly supervised datasets is much larger, the distant dataset includes almost 16 times more examples than the manually supervised one.

Using this distantly supervised dataset similar to the general domain experiments were performed with the supervised testing set for evaluation.

[14] https://metamap.nlm.nih.gov/

[15] https://www.ncbi.nlm.nih.gov/books/NBK9684/

4.3.5 Results

The results of the training of the network in various ways with distantly supervised data can be seen in the Table 4.10. For comparison, the supervised result also was included in the table.

Experiment	P	R	F1
Supervised training	**90.05**	82.14	**85.58**
Distantly supervised training	46.16	44.52	41.81
Distantly supervised + MIL	59.67	50.24	54.17
Distantly supervised + supervised data	76.81	67.62	70.45
Distantly supervised + supervised data + MIL	87.68	**83.33**	85.3
Transfer from supervised	54.07	48.57	51.17
Transfer from supervised + MIL	61.84	43.33	49.29

Table 4.10: Precision, Recall and F1-scores for distantly supervised training evaluation.

So it can be seen, that all the conclusions that were made for distantly supervised training in general domain, are also valid in the medical domain.

One interesting aspect of Rosario-Hearst dataset that it is highly unbalanced. The formed distantly supervised dataset is also unbalanced with the same relation "treatment-for" dominating over all others. It was interesting to see the effect of balancing of the dataset and the results of evaluation could be seen in the Table 4.11.

Experiment	P	R	F1
Supervised training	**90.05**	82.14	85.58
Distantly supervised training	43.1	35	37.83
Distantly supervised + MIL	56.22	60.24	55.87
Distantly supervised + supervised data	85.35	86.43	85.88
Distantly supervised + supervised data + MIL	84.17	**89.05**	**86.45**
Transfer from supervised	46.95	45.95	40.76
Transfer from supervised + MIL	60.14	43.57	50.53

Table 4.11: Precision, Recall and F1-scores for distantly supervised training evaluation with balanced distant dataset.

The results that were obtained by simply training on the balanced distant dataset are lower than for unbalanced one, but when the supervised data is mixed in the results become much higher. The first fact might be

explained by the basic requirement of any supervised learner: the training and the testing dataset are supposed to be drawn from the same distribution. But the second fact can either mean that distantly supervised data is always uncertain, so true distribution of classes cannot be seen just from a number of examples, or that simple neglecting of some of the dominating class examples helped the network to avoid noisy data. Also, one point that was observed while constructing the distant dataset is that the set of entity pairs for the relation "treatment-for" intersects with the set for "prevents-from" significantly. Thus, there are a lot of examples in the dataset, that will be labelled with both of the relations, that confuses the network more when all the examples are included (in unbalanced setup).

Representative trigrams The representative trigrams also give the same intuition that was there in the general domain - distantly supervised data helps to avoid the biased understanding of the relations. So for the supervised training for relation "prevents-from" they were:

vaccine for pneumonia, vaccination against influenza, vaccination against swine, vaccine against pneumococcus, polio vaccination participation, influenza vaccination on, influenza vaccination, hepatitis b vaccine, vaccine 6

They are clearly biased to various forms of vaccines and vaccinations. While for distantly supervised dataset mixed with supervised data were obtained:

malaria, nausea and vomiting, hiv-infected patients, tuberculosis, beclomethasone dipropionate, fluoride, cimetidine 1, ganciclovir

These ones are not human-interpretable, but they are definitely not biased to vaccinations. The amount of disease and drug names in the trigrams reveals that it was not easy for the network to concentrate on the syntactic constructions for the relation itself rather than on the names of the entities. This can be explained by the bad choice of "Other" class examples. As all the sentences with various named entities were used for it, only sentences containing exactly diseases and drugs were classified for the relations and the network started to overfit on the entity names. One possible way to fight this problem is to replace entity names with certain keywords in all the examples. The experiment revealed, that this modification of the dataset lead to the improvement of the score for distantly supervised training. The score became P=52.06, R=52.38, F1=51.54 against P=46.16, R=44.52, F1=41.81 for non-modified dataset. Representative trigrams also show a very good progress:

prevents-from: *entity1-resistant entity2, nausea and entity2, to prevent*

entity2, entity2, entity1 and entity2, use of entity2, entity1(entity2

treatment-for: *treatment of entity2, entity2, patients with entity2, treatment with entity1, treated with entity1, administration of entity1, response to entity1, entity2 with entity1*

But the result of the evaluation is lower than one for the non transformed dataset and with Multiple Instance Learning. The problem with this approach is that Multiple Instance Learning cannot be applied anymore in the previous form, as all the entity pairs are the same and the examples cannot be grouped to form the bags on the basement of different entity pairs. But some other approaches might be explored, for example, randomly grouping some fixed amount of examples for one class into bags.

Chapter 5

Conclusions

This chapter summarises the results of the conducted experiments and possible conclusions that can be observed. Furthermore, interesting ways for future development of the ideas and improvement of current results are discussed.

5.1 Conclusions

It can be concluded that a Convolutional Neural Network architecture can be successfully applied for the task of Information Extraction, or Relation Classification more precisely. In specific domains, such as the biomedical domain, it can outperform existing methods based on Natural Language Processing tools and manually extracted features.

The model configuration that showed the best performance for the general domain was not the same for the medical domain. Different types and lengths of embeddings, Word2Vec, Swivel and GloVe affect the performance in the setup of the considered problem. The initial quality of the embeddings influences the obtained results the most, e.g. Word2Vec embeddings trained on a large amount of data would perform better, while GloVe can capture latent features of words on smaller corpora. An interesting observation can be done from the validation experiments for the medical domain (see Subsection 4.2.4). It is that domain specific embeddings do result in better performance of the model comparing to the general embeddings, even when embeddings are trained during the network training.

Concerning Distant Supervision several main points should be noted:

- In the setup of real-world application for continuous extraction of new knowledge, Distant Supervision can be considered as the best approach with respect to overall effort among manual extraction and manually

supervised automated extraction. For example in the medical domain the amount of annually indexed articles on Medline is more than one million and it keeps growing every year [1]. In the case of Distant Supervision only the resulting sentences should be checked manually, while for supervised training the dataset should be created and also the results should be checked.

- The creation of the distantly supervised dataset does not require manual work of the experts but should be very carefully performed. The correspondence between the context of the textual corpora and required relational classes, the possible distribution of classes, the methods of aligning, the quality of the Knowledge Base should be taken into account.

- As the evaluation of the distantly supervised model is a quite challenging task, a way to inspect the training process should be introduced. The one considered in the thesis is extracting representative trigrams (see Subsection 3.3.1). They might even serve as a basis for future application of an Active Learning approach that can use representative trigrams for getting experts' feedback for improving the training dataset.

- There exist various ways of improving the Distant Supervision approach by mitigating the assumptions made by it. One of them, that was proven to show rather good results (see Subsection 4.3.2), is Multiple Instance Learning.

To summarise, the Ranking Convolutional Neural Network trained on distantly supervised dataset showed ability to perform Relation Extraction task both in the general domain and in the medical domain. Various ways to get an intuition about the knowledge learned by the network and to improve the results of the evaluation were investigated, among them representative trigrams and Multiple Instance Learning.

5.2 Future work

During the research, various ideas for future work that were not implemented due to the time constraints emerged. Among them:

- **Generation of the relational classes** One of the restrictions of the current model is a fixed schema of relations [Riedel et al., 2013]. It

[1] `http://dan.corlan.net/cgi-bin/medline-trend?Q=`

means that only a fixed set of relational classes can be learned and classified by the network. An alternative way is to generate relational classes names with a Recurrent Neural Network from a sentence representation obtained by the Ranking Convolutional Neural Network, thus letting various new relations being extracted.

- **Widening the extraction area** Sometimes relations can be described in several sentences, thus syntactical and semantical information from all of them is needed. To use it the training set can be formed from the whole paragraphs that include entities mentions.

- **Generative Adversarial Networks** Tackling the problem of the small amount of data for the training process might be also solved with Generative Adversarial Networks. The problem of text generation has already a lot of attention (e.g. [Hu et al., 2017]) and the results can be used as generators of the text for the Ranking Convolutional Neural Network that will classify relations.

- **Improving Multiple Instance Learning** As it is mentioned in [Lin et al., 2016] about Multiple Instance Learning *"It's apparent that the method will lose a large amount of rich information containing in neglected sentences."*. There are various different ways to improve it, for example, one of them is suggested by the authors of [Lin et al., 2016]. They implement selective attention mechanism for retrieving good examples.

- **Exploiting Representative Trigrams** Obtained during the training of the network representative trigrams might be used for the improvement of distantly supervised examples selection. Thus, if a sentence contains one of the trigrams it will be an example of the relation with higher probability. The other way of exploiting representative trigrams is to improve the quality of the distantly supervised dataset by filtering out sentences that contain not sensible trigrams (see Paragraph 4.3.3).

- **Improving preprocessing of the text** There are several ways to improve preprocessing, such as extracting n-grams, replacing entities names with placeholders (see Paragraph 4.1.4), improving the relational facts alignment in the Distant Supervision approach, shortening the length of the sentences in the training dataset (see Paragraph 4.3.3).

List of Figures

1.1 Example of Knowledge Graph 6

2.1 A framework for learning word vectors. Context of three words (the, cat, and sat) is used to predict the fourth word (on). The input words are mapped to columns of the matrix W to predict the output word. 10
2.2 12

3.1 Ranking Convolutional Neural Network 17
3.2 Multiple Instance Learning example 21
3.3 Schema of obtaining a trigram value 22
3.4 Tensorboard visualisation of the network 24

4.1 Experiments scheme 28
4.2 Semantic values, general domain supervised experiments . . . 38
4.3 Scores distributions, general domain supervised experiments . 39
4.4 Semantic values, medical domain supervised experiments . . . 47
4.5 Scores distributions, medical domain supervised experiments . 48
4.6 Relation of supervised and distantly supervised knowledge . . 52
4.7 Relation between correct answers for manually and distantly supervised training 53
4.8 Correlation of amount of correct and wrong answers with sentence length and distance between entities 54

List of Tables

2.1 Nearest neighbours for target words using GloVe vectors before and after counter-fitting 12

3.1 The word embeddings used for the experiments 14

4.1 Cross-validation for the general domain 32
4.2 Cross-validation for the general domain on locally trained embeddings 33
4.3 General domain supervised experiments results 33
4.4 Representative trigrams, SemEval dataset 35
4.5 Cross-validation for the medical domain 43
4.6 Medical domain, AIMed evaluation results 44
4.7 Medical domain, DDI evaluation results 44
4.8 Medical domain, Rosario-Hearst evaluation results 44
4.9 General domain, Distant Supervision experiments results . . . 50
4.10 Medical domain, Distant Supervision experiments results . . . 56
4.11 Medical domain, Distant Supervision with balancing experiments results 56

References

[Airola et al., 2008] Airola, A., Pyysalo, S., Björne, J., Pahikkala, T., Ginter, F., and Salakoski, T. (2008). All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning. *BMC bioinformatics*, 9(11):S2.

[Angeli et al., 2014] Angeli, G., Tibshirani, J., Wu, J., and Manning, C. D. (2014). Combining distant and partial supervision for relation extraction. In *EMNLP*, pages 1556–1567.

[Berger et al., 2000] Berger, A., Caruana, R., Cohn, D., Freitag, D., and Mittal, V. (2000). Bridging the lexical chasm: statistical approaches to answer-finding. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 192–199. ACM.

[Bojanowski et al., 2016] Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016). Enriching word vectors with subword information. corr abs/1607.04606.

[Boling and Das, 2014] Boling, C. and Das, K. (2014). Semantic similarity of documents using latent semantic analysis. *2014 NCUR*.

[Bunescu et al., 2005] Bunescu, R., Ge, R., Kate, R. J., Marcotte, E. M., Mooney, R. J., Ramani, A. K., and Wong, Y. W. (2005). Comparative experiments on learning information extractors for proteins and their interactions. *Artificial intelligence in medicine*, 33(2):139–155.

[Bunescu and Mooney, 2005] Bunescu, R. and Mooney, R. J. (2005). Subsequence kernels for relation extraction. In *NIPS*, pages 171–178.

[Craven et al., 1999] Craven, M., Kumlien, J., et al. (1999). Constructing biological knowledge bases by extracting information from text sources. In *ISMB*, volume 1999, pages 77–86.

[Dietterich et al., 1997] Dietterich, T. G., Lathrop, R. H., and Lozano-Pérez, T. (1997). Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 89(1):31–71.

[dos Santos et al., 2015] dos Santos, C. N., Xiang, B., and Zhou, B. (2015). Classifying relations by ranking with convolutional neural networks. *CoRR*, abs/1504.06580.

[Dumais, 2004] Dumais, S. T. (2004). Latent semantic analysis. *Annual review of information science and technology*, 38(1):188–230.

[Ghosh et al., 2016] Ghosh, S., Chakraborty, P., Cohn, E., Brownstein, J. S., and Ramakrishnan, N. (2016). Characterizing diseases from unstructured text: A vocabulary driven word2vec approach. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 1129–1138. ACM.

[Gormley et al., 2015] Gormley, M. R., Yu, M., and Dredze, M. (2015). Improved relation extraction with feature-rich compositional embedding models. *arXiv preprint arXiv:1505.02419*.

[Hendrickx et al., 2009] Hendrickx, I., Kim, S. N., Kozareva, Z., Nakov, P., Ó Séaghdha, D., Padó, S., Pennacchiotti, M., Romano, L., and Szpakowicz, S. (2009). 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.

[Hu et al., 2017] Hu, Z., Yang, Z., Liang, X., Salakhutdinov, R., and Xing, E. P. (2017). Controllable text generation. *arXiv preprint arXiv:1703.00955*.

[Jśnior et al., 2017] Jśnior, E. A. C., Marinho, V. Q., and dos Santos, L. B. (2017). Nilc-usp at semeval-2017 task 4: A multi-view ensemble for twitter sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 611–615.

[Le and Mikolov, 2014] Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1188–1196.

[Levy and Goldberg, 2014] Levy, O. and Goldberg, Y. (2014). Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of*

the Association for Computational Linguistics (Volume 2: Short Papers), volume 2, pages 302–308.

[Lin et al., 2016] Lin, Y., Shen, S., Liu, Z., Luan, H., and Sun, M. (2016). Neural relation extraction with selective attention over instances. In *Proceedings of ACL*, volume 1, pages 2124–2133.

[Mikolov et al., 2013a] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[Mikolov et al., 2013b] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

[Mintz et al., 2009] Mintz, M., Bills, S., Snow, R., and Jurafsky, D. (2009). Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, ACL '09, pages 1003–1011, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Mitra et al., 2016] Mitra, B., Nalisnick, E., Craswell, N., and Caruana, R. (2016). A dual embedding space model for document ranking. *arXiv preprint arXiv:1602.01137*.

[Mohammad et al., 2008] Mohammad, S., Dorr, B., and Hirst, G. (2008). Computing word-pair antonymy. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 982–991. Association for Computational Linguistics.

[Mrkšić et al., 2016] Mrkšić, N., Ó Séaghdha, D., Thomson, B., Gašić, M., Rojas-Barahona, L., Su, P.-H., Vandyke, D., Wen, T.-H., and Young, S. (2016). Counter-fitting word vectors to linguistic constraints. In *Proceedings of HLT-NAACL*.

[Nguyen and Grishman, 2015] Nguyen, T. H. and Grishman, R. (2015). Combining neural networks and log-linear models to improve relation extraction. *arXiv preprint arXiv:1511.05926*.

[Pennington et al., 2014] Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

[Ramon and De Raedt, 2000] Ramon, J. and De Raedt, L. (2000). Multi instance neural networks. In *Proceedings of the ICML-2000 workshop on attribute-value and relational learning*, pages 53–60.

[Ramos et al., 2003] Ramos, J. et al. (2003). Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 133–142.

[Riedel et al., 2010] Riedel, S., Yao, L., and McCallum, A. (2010). Modeling relations and their mentions without labeled text. *Machine learning and knowledge discovery in databases*, pages 148–163.

[Riedel et al., 2013] Riedel, S., Yao, L., McCallum, A., and Marlin, B. M. (2013). Relation extraction with matrix factorization and universal schemas.

[Robertson et al., 2009] Robertson, S., Zaragoza, H., et al. (2009). The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

[Roller and Stevenson, 2014] Roller, R. and Stevenson, M. (2014). Applying umls for distantly supervised relation detection. In *Proceedings of the 5th International Workshop on Health Text Mining and Information Analysis (Louhi)@ EACL*, pages 80–84. Citeseer.

[Rosario and Hearst, 2004] Rosario, B. and Hearst, M. A. (2004). Classifying semantic relations in bioscience texts. In *Proceedings of the 42nd annual meeting on association for computational linguistics*, page 430. Association for Computational Linguistics.

[Segura Bedmar et al., 2011] Segura Bedmar, I., Martinez, P., and Sánchez Cisneros, D. (2011). The 1st ddiextraction-2011 challenge task: Extraction of drug-drug interactions from biomedical texts.

[Shazeer et al., 2016a] Shazeer, N., Doherty, R., Evans, C., and Waterson, C. (2016a). Swivel: Improving embeddings by noticing what's missing. *arXiv preprint arXiv:1602.02215*.

[Shazeer et al., 2016b] Shazeer, N., Doherty, R., Evans, C., and Waterson, C. (2016b). Swivel: Improving embeddings by noticing what's missing. *CoRR*, abs/1602.02215.

[Vu et al., 2016] Vu, N. T., Adel, H., Gupta, P., and Schütze, H. (2016). Combining recurrent and convolutional neural networks for relation classification. *arXiv preprint arXiv:1605.07333*.

[Zeng et al., 2015] Zeng, D., Liu, K., Chen, Y., and Zhao, J. (2015). Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP), Lisbon, Portugal*, pages 17–21.

[Zeng et al., 2014] Zeng, D., Liu, K., Lai, S., Zhou, G., Zhao, J., et al. (2014). Relation classification via convolutional deep neural network. In *COLING*, pages 2335–2344.

[Zhang and Wang, 2015] Zhang, D. and Wang, D. (2015). Relation classification via recurrent neural network. *CoRR*, abs/1508.01006.