Gesellschaft für Informatik (Hrsg.): SKILL 2019,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2019 

# Social Relation Extraction from Chatbot Conversations

### A Shortest Dependency Path Approach

Markus Glas[1]

**Abstract:** Digital dialog systems, also known as chatbots, often lack in the sense of a human-like and individualized interaction. The ability to learn someone's social relations during conversations can lead to more personal responses and therefore to a more human-like and diverse conversation. In this work we present S-REX, a comparison method for extracting social relations from chatbot conversations. The implemented approach uses information from the shortest dependency path in combination with state-of-the-art natural language processing models for entity recognition and semantic word vectors. The method is evaluated on two conversational datasets and achieves results close to more complex neural network methods without the need of extensive training.

**Keywords:** Relation Extraction; Information Extraction; Chatbots; Information Retrieval; Text Mining; Natural Language Processing

## 1 Introduction

Digital dialog systems, also known as chatbots, have gained popularity through their usage in smart phones and smart speakers in recent years. Nevertheless, conversations with chatbots often lack in terms of natural human-like conversations and are often based on question and answering dialogs. The ability to learn someone's social relationships during conversations can help to create personalized responses and lead to a more human-like and diverse conversation.

The field of relation extraction (RE), as a subfield of information extraction (IE), has become an important task to automatically discern relationships among detected entities within unstructured text. Within the field of information extraction, several applications, most notably for optimizing web search engines, have pushed the development and research of new approaches during the last two decades. Nevertheless, most RE approaches have essentially focused on textual data in form of books or web site documents. A less considered area is the field of natural language conversations, used within chat messages between people, or humans and chatbots. The ambiguous, informal and often erroneous characteristic of natural language conversations requires specific adaptions on common approaches in

[1] Technische Hochschule Nürnberg Georg Simon Ohm, Fakultät Informatik, Hohfederstraße 40, 90489 Nürnberg, Germany, glasma60952@th-nuernberg.de

RE. Therefore, it is necessary to evaluate new techniques upon the well-known approaches for extracting relationships. This work evaluates current approaches in RE and their usage within the domain of natural language conversations. Based on that, we present a new approach, called *S-REX*, for extracting social relations within chat conversations.

## 2 Relation Extraction

Relation extraction (RE), also known as relationship extraction or relation detection, describes the task of detecting and classifying semantic relations among named entities in texts [JM14]. These are commonly represented as binary relations between two entities such as *part-of*, *child-of*, *employed-at* or *located-in*. Relation extraction faces many challenges as relations are inherently ambiguous and the range of relations can vary from one domain to another. As other computational linguistic tasks, RE is also language-dependant.

Approaches for semantic relation extraction can be roughly assigned to one of the following four categories [Ba16, JM14]: Pattern-based, Supervised, Semi-Supervised or Distant Supervision, which are being discussed throughout the following sections.

### 2.1 Pattern-based Relation Extraction

Early and some of the commonly used algorithms use hand-written patterns for relation extraction, also known as pattern-based RE. First developed by Hearst [He92], these patterns try to infer the relationships between entities based on syntactic rules using regular expressions or similar. Consider the following sentence from Hearst's publication:

*"The bow lute, such as Bambara ndang, is plucked and has individual curved neck for each string."*

Even without knowing the meaning of *Bambara ndang*, a human can infer that it is a kind of *bow lute*. This is true, even if the reader has only few conception about a *bow lute*.

A lexico-syntactic pattern like $NP_0$ *such as* $\{NP_1,\ NP_2\ ..., (and|or)\}\ NP_n$ can indicate this relation by recognizing noun phrase (NP) followed by the words *such as*, and one or more noun phrases, connected by *and* or *or*. Noun phrases are commonly composed of a noun (NN), with an optional preceding determiner (DT) and adjective (JJ). A regular expression for extracting noun phrases, based on part-of-speech (POS) tags, could be defined as follows: *<DT>?<JJ>*<NN>*.

Even if this method can reach a high precision on specific domains, it requires deep domain knowledge and lacks with respect to robustness and portability to other domains. Nevertheless, it is still a commonly used approach in some enterprise systems [CLR13].

### 2.2 Supervised Relation Extraction

Supervised relation extraction, based on supervised machine learning, follows the method of hand-annotating a text with a fixed set of relations as training corpus, and use them to train a classifier. In recent approaches, two classifiers are used to indicate if two entities are related to each other or not, before the final decision about the entity type is made. This allows to speed up the final classification and allows the use of distinct features. For each of the classifiers, standard classification techniques like Support Vector Machines, logistic regression, decision trees or similar, can be used [JM14].

Supervised relation extraction can achieve high accuracy, if enough hand-labelled data is available. But labelling large training sets is extremely time consuming and error-prone. Besides that, supervised models are very domain-specific and can not generalize to different text genres, that were not part of the training set [JM14]. Most research has therefore focused on semi-supervised approaches in recent years.

### 2.3 Semi-supervised Relation Extraction

Semi-supervised or bootstrapping approaches for RE try to tackle the problem of hand-labelling large training sets. Starting with a set of known relationships, bootstrapping algorithms try to generate training data by iterating over large texts and extracting training sentences based on predefined seeds.

One of the first published system using this technique was DIPRE, developed by Brin [Br98] specifically for the purpose of extracting books and their authors from web site documents (i.e. *[author, book]* pairs). It uses a 7 tuple <*author, book, order, url, prefix, suffix, middle*> as feature for every co-occurrence of a seed example, where *order* represents a boolean if the author occurs before the book, *prefix* and *suffix* the 10 words before and after the matched entities and *middle* all words in between them. All tuples are grouped by the *middle* and *order* features and DIPRE verifies if new tuples are the same. The new learned pattern is used to search the corpus again and extract new relationships.

Agichtein & Gravano [AG00] extended Brin's approach to a system called *Snowball*, which presents a more general approach for extracting further entity types from texts, than only books and authors. Similar to DIPRE, this approach forms tuples of the entities (e1, e2) and the words before (BEF), in between (BET) and after (AFT) them as context tuple: <*BEF, e1, BET, e2, AFT*>. Each context tuple is then represented by a TF-IDF vector, and similar contexts are clustered by a single-pass clustering, using the cosine similarity between the vectors.

Batista et al. [BMS15] updated the *Snowball* system with state-of-the-art word vector representations using word embeddings. Together with a prior selection of words between the entities using hand-written patterns, this approach achieves a higher accuracy compared to previous systems.

The advantage of bootstrapping approaches lies in the omitted necessity of hand labelling large datasets to train a classifier. One downside of the examined bootstrapping approaches is the often limited usability to specific relation types, such as hypernyms (is-a) or meronyms (part-of) [GBM03, SJN05]. Another drawback of semi-supervised RE is the manual creation of seed examples, which requires domain specific knowledge.

### 2.4 Distant Supervision

Distant Supervision methods use knowledge bases, like *DBPedia*[2], *Wikidata*[3] or formerly *Freebase*[4] (offline since May 2016), containing semantic relations to automatically collect large amounts of seed examples. These seed examples are then used to extract training examples on large amounts of texts, similar to bootstrapping approaches.

Considering a system which tries to learn the relation type *place-of-birth*, instead of manually defining a small number of seed examples, *DBPedia* is used, containing over 100,000 examples for the relation type *place-of-birth*, like <*Albert Einstein, Ulm*>. Distantly supervised approaches then run a named entity tagger over large coherent texts, like Wikipedia Articles, and extract sentences containing both entities (e.g. *"Albert Einstein was born in Ulm..."*). The extracted sentences can be used to train a supervised classifier, similar to the supervised approaches described above.

Mintz et al. [Mi09] use *Freebase* as source for their seeds, which is a former semantic database trained on Wikipedia info boxes and other sources. They use several syntactic and lexical features, like sequence of words between the entities and part-of-speech tags, to train a multi-class logistic classifier.

Aljamel et al. [AOA15] used a combination of *DBPedia* and *Freebase* for extracting their seed tuples. They also used *JENA'S SPARQL* engine for extracting the mentioned relations within both databases through a common interface. For the classification task, they compared three methods: Support Vector Machine (SVM), Perceptron Algorithm Uneven Margin (PAUN) and K-Nearest-Neighbour (KNN). The SVM outperformed in both Person-Location (Per-Loc), Person-Organisation (Per-Org) relations and had an equal accuracy to PAUN for Location-Organisation (Loc-Org) relations.

[2] https://wiki.dbpedia.org/
[3] https://www.wikidata.org
[4] https://developers.google.com/freebase/

# 3 Social Relation Extraction

Social Relations describe relationships between two or more persons. Within this work, social relations are denoted as a finite number of relation types between two entities within a sentence. The entities are assigned to one of the following two categories: *Person (PER)* or *User (USR)*. The type *Person (PER)* describes a person, recognized by a named entity recognition (NER) model. The entity type *User (USR)* describes the user interacting with the dialog system. This applies when the user mentions him- or herself through a possessive pronoun, like *my* or *I* (e.g. "*<USR>*My*</USR>* brother *<PER>*Michael*</PER>* is flying to London next week.".

Entity pairs must occur in one of the following orders within a sentence, to be successfully extracted: *Person-to-Person* (PER-PER) or *User-to-Person* (USR-PER). Table 1 shows an example for both combinations.

| No. | Example | Relations | Entities |
|---|---|---|---|
| 1 | "**Peter** is the father of **Tom**." | <Peter, father-of, Tom> | PER-PER |
| 2 | "**My** daughter **Lisa** is moving to London next month." | <Lisa, daughter-of, USER> | USR-PER |

Tab. 1: Social relationships are being described as a relation between two specific types of entities, which can be related to each other in two different ways shown as rows in the table.

The relations between the two entities can be of one of the following types: *father-of*, *mother-of*, *son-of*, *daughter-of*, *brother-of*, *sister-of*, *grandfather-of*, *grandmother-of*, *grandson-of*, *granddaughter-of*, *husband-of*, *wife-of*, *uncle-of*, *aunt-of* or *friend-of*.

Extracted relations are denoted as a triple $< e_1, rel, e_2 >$, where $e_1$ and $e_2$ represent the extracted entities, and $rel$ describes a specific relationship type that connects the two entities.

## 3.1 Shortest Dependency Path Relation Extraction Approach

The method for shortest path RE is based on the feature extraction process, used by Bunescu et al. [BM05] to train a multi-class classifier. Other than in [BM05], the proposed method does not train a classifier due to the lack of conversational training datasets. It will rather use a distance measurement and compare the extracted features with predefined vectors of known relationship types. Besides that, it uses word embeddings instead of combined features made of POS-tags, entity types and other syntactic features.

## 3.2 Shortest Dependency Path Hypothesis

As proposed by Bunescu et al. [BM05], the contribution of the dependency path for establishing the relation between two entities, is almost exclusively concentrated in the shortest path between these in the undirected version of the dependency path. If two entities are arguments of the same predicate (verb) and belong to the same predicate-argument structure, then the shortest path will pass through this predicate. Figure 1 shows an example representing this case, where the two entities represent the subject (nsubj - *protesters*) and object (dobj - *stations*), both related to the same predicate.

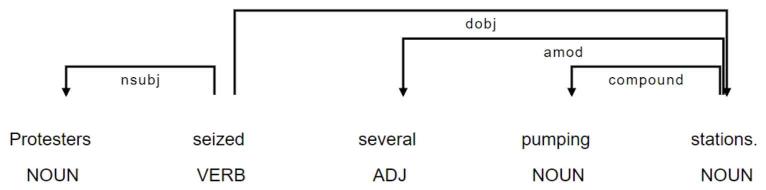


Fig. 1: Sentences with POS-Tags and dependency path

| Sentence | Shortest path in undirected dependency graph |
|---|---|
| **Protesters** seized several pumping **stations**. | **protesters** ← *seized* → **stations** |

Tab. 2: Shortest path representations of the sentence

Within the sentence shown in figure 1, the shortest dependency path (SDP) between the two noun entities *protesters* and *stations* passes through the predicate (verb) *seized*. In this basic examples, the predicate represents the only word on the shortest path, which might not be the case in more complex sentences. Nevertheless, even with more words on the shortest path, the predicate is always a part of it.

### 3.3 Shortest Path Hypothesis for Social Relations

The shortest path hypothesis can be used to extract social relationships between two entities, as the shortest path commonly passes through the mentioned relation. This is also true, if the relation is not mentioned between two entities or if other entities appear in between them. The sentence shown in the figure 2, holds a social relation between the entities *Monica* and *Ross*. Even though the relation is not in between these entities, it is possible to extract the relation by following the shortest path in the undirected dependency graph.

### 3.4 S-REX - Social Relationship Extraction

S-REX combines steps from different RE techniques, as described in section 2, and consists of eight steps: Recognize entities, extract dependency path, transform dependency into an undirected graph, search shortest path, get word embeddings for each word, sum up word embeddings, measure cosine similarity, extract relationships. For visualization purposes the

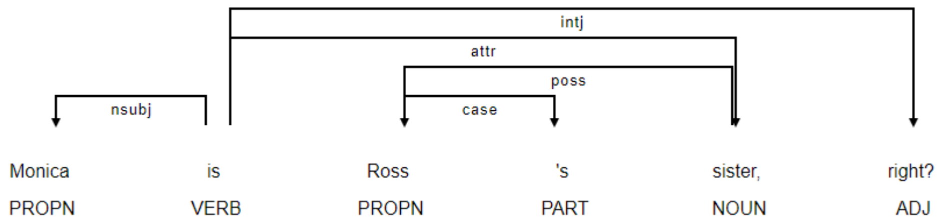


Fig. 2: Sentence containing social relations with POS-Tags and dependency path

sentence *"Monica is Ross's sister, right?"*, as illustrated in figure 2, is used as an example throughout this section.

The first step consists of finding entities (*PER*, *USR*) within a sentence. *Person (PER)* entities are recognized by using a pre-trained sequence tagger model, provided by the open source library *Flair*[5]. The model is trained over the *CoNLL-03* dataset and offers a 4-class named entity recognition. The architecture of the sequence labelling follows a BiLSTM-CRF approach, as described in [ABV18]. *USER (USR)* entities are recognized by using a predefined list.

For creation of the dependency path (step 2), the pre-trained model *en_core_web_md* for English, provided by the open source NLP library *spaCy*[6], is used. The model is trained on the sources *CommonCraw*[7] and *OneNotes 5*[8], and consists of 20,000 unique word vectors and 685,000 keys.

The extracted dependency path is then transformed into an undirected graph containing the words as nodes and dependencies as edges (see Figure 3). As the entities are already known from the first step, the next step is to search for the shortest path between each entity pair using Dijkstra's algorithm. The entity pairs are constructed, following their appearance within a sentence from left to right, so that the leftmost entity is connected to each of following entities to the right. For instance, considering three entities $e_1, e_2, e_3$ within a sentence, the three entity pairs $< e_1 - e_2 >$, $< e_1 - e_3 >$ and $< e_2 - e_3 >$ are being constructed.

Following the sentence of the example, two named entities are recognized, resulting in one entity pair $< Monica - Ross >$. Using the shortest path algorithm, the words *is* and *sister* are extracted (step 4; see Table 3).

In the fifth step, for each word on the shortest path the corresponding word embedding representation is looked-up in a word embedding dictionary. The dictionary used here,

[5] https://github.com/zalandoresearch/flair
[6] https://spacy.io/
[7] http://commoncrawl.org/
[8] https://catalog.ldc.upenn.edu/LDC2013T19

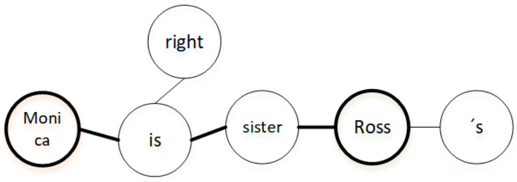


Fig. 3: Example of a dependency path within a sentence containing two entities.

| Sentence | Shortest path in undirected dependency graph |
|---|---|
| **Monica** is **Ross**'s sister, right? | **Monica** $\leftarrow is \rightarrow sister \rightarrow$ **Ross** |

Tab. 3: Shortest path representations of the sentence

contains vector representations of *GloVe*[9] word embeddings, trained over Wikipedia provided by the *FastText*[10] library. Each word is represented by a 100-dimensional word vector. Words which are not part of the dictionary, will be represented by a zero vector.

All word vectors of the shortest path are summed up into a single 100-dimensional vector (step 6), further denoted as *shortest path vector* ($\vec{sp}$). This vector is compared (step 7) to a set of stored *relation vectors* ($\vec{rel}$), each representing one of the predefined relation types. This results in a word vector for each of the following 15 words: *father*, *mother*, *sister*, *brother*, *son*, *daughter*, *husband*, *wife*, *grandson*, *granddaughter*, *grandmother*, *grandfather*, *uncle*, *aunt* and *friend*.

The *relation vector* with the highest cosine similarity, defines the relation type between the entities (step 8). The similarity is depicted by a single float value and needs to exceed a predefined distance threshold of 0.6 for successful RE. Similarities below this threshold are not extracted as social relationships. The similarity is measured by using the cosine similarity, as denoted in Equation 1, where $\vec{sp}$ represents the shortest path vector and $\vec{rel}(n)$ a relation vector ($1 \leq n \leq 15$).

$$similarity = cos(\theta) = \frac{\vec{sp} \cdot \vec{rel}(n)}{||\vec{sp}|| \cdot ||\vec{rel}(n)||} \tag{1}$$

Figure 4 illustrates a two-dimensional plot of *relation vectors* (blue) and the *shortest path vector* (red) between the entities *Monica* and *Ross* from the example. The closest vector in this case is the word vector for representing *sister*.

[9] https://nlp.stanford.edu/projects/glove/
[10] https://fasttext.cc/docs/en/crawl-vectors.html

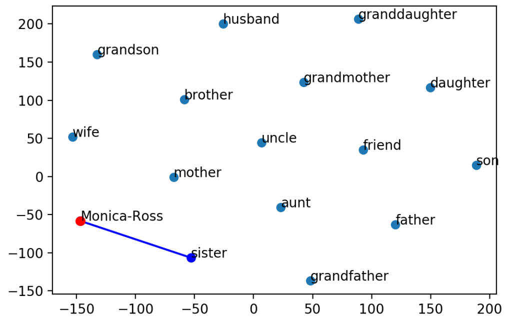


Fig. 4: Word Vector Space of 100-dimensional word embeddings projected to a 2-dimensional space using t-Distributed Stochastic Neighbour Embedding (t-SNE). The closest distance from vector *Monica-Ross* is the vector for representing the word *sister*.

## 4 Experimental Evaluation

The experiments are conducted on a dataset, built as a combination from utterances of the *Persona-Chat*[11] dataset and the *Friends TV Corpus*[12].

### 4.1 Datasets

The *Persona-Chat* project [Zh18] was created by collecting 162,064 utterances between crowdworkers via *Amazon Mechanical Turk*. The crowdworkers were randomly paired and told to chat naturally and get to know each other during conversation. This dataset is also known as a training set for the Conversational Intelligence Challenge 2 (ConvAI2).

The *Friends TV Corpus*, first introduced by Nio et al. [Ni14], was constructed by using subtitles of the popular American *Friends*[13] sitcom. It contains subtitles from five seasons, with a total of 112 episodes. Each episode contains several scenes with several dialog turns. The dataset consists of 60,849 dialogs in total.

Both datasets contain sentences in a natural non-upscale language. The *Persona-Chat* corpus mainly contains *chat-style* utterances with an unclean spelling, including typing errors, chat-slang notations and missing punctuation. The *Friends TV Corpus* contains less spelling and grammatical errors, but, different from the *Persona-Chat* corpus, also contains mentions of persons. To evaluate the extraction of all types of defined entities (*PER* and

[11] https://github.com/facebookresearch/ParlAI/tree/master/parlai/tasks/convai2
[12] https://github.com/npow/friends-chatbot
[13] https://en.wikipedia.org/wiki/Friends

*USR*), a combination of both datasets is used containing 500 randomly chosen utterances from each dataset. The combined dataset consists of 2313 sentences, which is an appropriate amount compared to other relation extraction datasets, such as the popular *SemEval-2010 Task 8* [He10], which has 2717 utterances in its test set.

### 4.2 Results

The 2313 sentences of the dataset contain 47 sentences with social relations. Within this category, seven sentences returned no relation and 4 returned a false or incomplete relation by the relation extraction system. This results in 11 false and 36 correct extracted sentences resulting in an accuracy of 76.6% and an F1 score of 79.6.

Sentences like *"my dad flies airplanes."* or *"That's what my mom said."* could be correctly extracted, even though the relation types *dad* and *mom* are not explicitly defined inside the relationship list. This is possible due to the use of semantic word vectors. Other sentences like *"I exercise at home with my gf"* or *"i actually have four sisses too !"* could not be extracted, due to unrecognized short notation or the use of slang. Besides that, the named entity recognizer might not recognize every person as such and sometimes falsely identifies other words as a person.

### 4.3 Comparison with other methods

An exact comparison with other methods is not possible, as most academic research on relation extraction focuses on general relation extraction tasks using supervised methods, trained on annotated corpora. State-of-the-art approaches use different methods including distant supervision, dependency path or end-to-end models. We compare our model with other models, trained on the *SemEval-2010 Task 8* [He10] as it is one of the most popular in current research. Current state-of-the-art results are achieved by Wu et al. [WH19] with an F1 of 89.25 by leveraging a pre-trained end-to-end model from the BERT [De18] language model. Cai et al. [CZW16] use a combination of convolutional neural networks with a two-channel recurrent neural network using long-short term memory (LSTM) and reach an F1 of 86.3. Xu et al. [Xu16] use deep recurrent neural networks for relation classification with a F1 of 86.1. Both of the latter approaches make use of information along the SDP.

All of the mentioned supervised neural network approaches reach a F1 score of over 86 and outperform S-REX. However, it should be noted that S-REX does not train a model and therefore can be instantly used without the need of a large annotated training set. Beyond that, it is evaluated on a different training set including different (only social) relation types.

## 5 Conclusion

In this work we present S-REX, a method for extracting social relationships from natural language conversations. The proposed method shows robustness in extracting social relations from conversational data, even if the grammatical structure is not correct or spelling errors are present. Additionally, different notations of relations can be extracted, due to the use of semantic word vectors. S-REX is also able to extract relations mentioned after two entities within a sentence. In future work, the presented method can be improved by expanding the number of features and taking relations across sentence boundaries into account, using coreference resolution.

## References

[ABV18] Akbik, A.; Blythe, D.; Vollgraf, R.: Contextual String Embeddings for Sequence Labeling. In: Proceedings of the 27th International Conference on Computational Linguistics. Association for Computational Linguistics, Santa Fe, New Mexico, USA, pp. 1638–1649, August 2018.

[AG00] Agichtein, E.; Gravano, L.: Snowball: extracting relations from large plain-text collections. In: Proceedings of the fifth ACM conference on Digital libraries - DL '00. ACM Press, San Antonio, Texas, United States, pp. 85–94, 2000.

[AOA15] Aljamel, A.; Osman, T.; Acampora, G.: Domain-Specific Relation Extraction - Using Distant Supervision Machine Learning:. In: Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management. SCITEPRESS - Science and and Technology Publications, Lisbon, Portugal, pp. 92–103, 2015.

[Ba16] Batista, D. S.: Large-Scale Semantic Relationship Extraction for Information Discovery. PhD Thesis, INSTITUTO SUPERIOR TÉCNICO, 2016.

[BM05] Bunescu, R. C.; Mooney, R. J.: A Shortest Path Dependency Kernel for Relation Extraction. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing. HLT '05, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 724–731, 2005.

[BMS15] Batista, D. S.; Martins, B.; Silva, M. J.: Semi-Supervised Bootstrapping of Relationship Extractors with Distributional Semantics. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Lisbon, Portugal, pp. 499–504, September 2015.

[Br98] Brin, S.: Extracting Patterns and Relations from the World Wide Web. In: WebDB. pp. 172–183, 1998.

[CLR13] Chiticariu, L.; Li, Y.; Reiss, F. R.: Rule-Based Information Extraction is Dead! Long Live Rule-Based Information Extraction Systems! In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Seattle, Washington, USA, pp. 827–832, October 2013.

[CZW16] Cai, R.; Zhang, X.; Wang, H.: Bidirectional Recurrent Convolutional Neural Network for Relation Classification. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Berlin, Germany, pp. 756–765, 2016.

[De18] Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs], October 2018. arXiv: 1810.04805.

[GBM03] Girju, R.; Badulescu, A.; Moldovan, D.: Learning semantic constraints for the automatic discovery of part-whole relations. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1. Association for Computational Linguistics, pp. 1–8, 2003.

[He92] Hearst, M. A.: Automatic Acquisition of Hyponyms from Large Text Corpora. In: COLING 1992 Volume 2: The 15th International Conference on Computational Linguistics. Association for Computational Linguistics, 1992.

[He10] Hendrickx, I.; Kim, S. N.; Kozareva, Z.; Nakov, P.; Seaghdha, D. O.; Pado, S.; Pennacchiotti, M.; Romano, L.; Szpakowicz, S.: SemEval-2010 Task 8: Multi-way Classification of Semantic Relations Between Pairs of Nominals. In: Proceedings of the 5th International Workshop on Semantic Evaluation. SemEval '10, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 33–38, 2010. event-place: Los Angeles, California.

[JM14] Jurafsky, D.; Martin, J. H.: Speech and language processing, volume 3. Pearson Education International, Upper Saddle River, NJ, 2 edition, 2014.

[Mi09] Mintz, M.; Bills, S.; Snow, R.; Jurafsky, D.: 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 - ACL-IJCNLP '09. volume 2, Association for Computational Linguistics, Suntec, Singapore, p. 1003, 2009.

[Ni14] Nio, L.; Sakti, S.; Neubig, G.; Toda, T.; Adriani, M.; Nakamura, S.: Developing non-goal dialog system based on examples of drama television. In: Natural Interaction with Robots, Knowbots and Smartphones, pp. 355–361. Springer, 2014.

[SJN05] Snow, R.; Jurafsky, D.; Ng, A. Y.: Learning syntactic patterns for automatic hypernym discovery. In: Advances in neural information processing systems. pp. 1297–1304, 2005.

[WH19] Wu, S.; He, Y.: Enriching Pre-trained Language Model with Entity Information for Relation Classification. CoRR, abs/1905.08284, 2019.

[Xu16] Xu, Y.; Jia, R.; Mou, L.; Li, G.; Chen, Y.; Lu, Y.; Jin, Z.: Improved Relation Classification by Deep Recurrent Neural Networks with Data Augmentation. arXiv:1601.03651 [cs], January 2016. arXiv: 1601.03651.

[Zh18] Zhang, S.; Dinan, E.; Urbanek, J.; Szlam, A.; Kiela, D.; Weston, J.: Personalizing Dialogue Agents: I have a dog, do you have pets too? arXiv:1801.07243 [cs], January 2018.