FRI Natural language processing course 2022
Project report

University *of Ljubljana*
Faculty *of Computer and*
Information Science

Automatic semantic relation extraction

Aljoša Koren, Klemen Škrlj, and Tilen Kavčič

Abstract

Keywords
Semantic relation extraction, TermFrame knowledge base

Advisors: Slavko Žitnik

1. Introduction

Various natural language processing techniques have been proposed to tackle relation extraction. These techniques are most often shown on broad language corpora like the New York Times corpus [1] and are trained and tested in the most common languages, most often English. In this paper we focus on the Karstology domain, with sentences in Slovenian, English and Croatian. Sentences used for model training are hand annotated by linguists. We explore how we can use current natural language techniques to extract both hypernym—hyponym pairs and more non-hierarchical semantic relations. Since our goal is that our methods not only work on this very domain-specific corpora, we also test our models on the SemEval2010 Task-8 dataset [2].

2. Related work

There are usually two types of algorithms for discovering the hyponym-hypernym relation; pattern-based and distributional methods. The pattern-based approach is usually time-consuming and language dependent, even if we take the same language from a different time period. Distributional methods can be supervised or unsupervised. They use word distribution to extract hypernyms. Roller et al. [3] compared both approaches in 2018. Both approaches use co-occurrences within a context, however pattern-based use predefined manually chosen patterns, while distributional methods use unconstrained word co-occurrences. They have extracted simple Hearst patterns and also broader patterns, took frequency of occurrences and sparsity into account and postpreprocessing which removed pairs that did not occur in enough sentences. This method was discovered to be better then the rest distributional methods they were comparing it with. The work done by Atzori and Balloccu [4] used a unsupervised learning for hypernym discovery. They used cosine distance in vector word embeddings as it was done before, but they added rank weighted by word frequencies in a corpus and level of similarity, to remove the semantic relations that might not be in the hyponym-hypernym relation. Their system is domain and language independent, because they do not use Hearst patterns [5] (realtions of the form x is-a y) or stopwords, but solely unstructured data.

Paper [6] describes results of a challenge where competitors had to extract hypernym-hyponym relations between a given list of domain-specific terms. The best approach was based on Hearst patterns where the team made use of large web corpus. The distributional approaches got competitive recall but struggled with precision. In a competition [7] teams were presented with a task of finding suitable hypernyms of a given word from the target corpus. The corpuses were multilingual and also domain-specific. Here, a system that learned embeddings of hyponym-hypernym pairs combined with unsupervised learning of Hearst-style patterns performed the best. Overall, supervised methods showed clear superiority over unsupervised ones.

Dependency trees are commonly used to tackle extraction of new intra-sentence instances of semantic relations. To keep only the most relevant information pruning is used. Zhijiang Guo , Yan Zhang and Wei Lu (2019) [8], for example, introduce Attention Guided Graph Convolutional Networks s (AGGCNs) which use a soft-pruning approach to clean up the dependency tree and keep the relevant sub-structures useful for the relation extraction task. AGGCNs transform the dependency tree into a fully connected edge-weighted graph. Weights represent the relatedness between nodes. They achive 69% F-score in TACRED dataset on sentence level relation classification. Since dependency trees don't capture inter-sentence relations, recurrent neural networks (RNNs) and convolutional neural networks (CNNs) are often used. Sahu et al. 2019 [9] employ their edge Graph CNN (GCNN) model to capture local and non-local dependencies. The graph nodes represent words while edges correspond to semantic dependencies. They use MIL-based bi-affine pairwise scoring function to infer relations between entities from the entity node representations.

BERT-based pre-training model with cascade tagging framework was first used by Kang Zhaoab et al. [10]. They used their cascade binary tagging framework in which in order to solve the problem of multiple overlapping triplets present in the same sentence share the same entities. When employing a pre-trained BERT encoder their model achieved F1 score of 89.6 on the NYT dataset and 91.8 on the WebNLG dataset.

Kang Zhaoab et al. [11] build on this work by introducing graph neural networks. They represent the words as nodes on the graphs and iteratively fuse two types of semantic nodes using the message passing mechanism. With this method they managed to improve the F1 score on the NYT dataset to 92.0 and WebNLB to 92.6. Their model managed to perform on par to the state-of-the-art on the SemEval-2010 Task 8, with the F1 score of 91.3.

The state-of-the-art on this relation extraction task is the QA model by Amir DN Cohen, Shachar Rosenman and Yoav Goldberg [12]. They approach the task as a span-prediction problem, which besides on the SemEval-2010 Task 8 also achieves state-of-the-art results on the TACRED dataset. Instead of formulating the problem as a relation classification task, where a sample encompasses a sentence, two head and tail entities and relation. Span Prediction, however formulates the problem using only a sentence, query and an answer to the query. They then use BERT to achieve F1 score of 91.9 on the SemEval-2010 Task 8.

3. Methods

As for the first method we implemented relation extraction with BERT [13]. We focused on the genus, location and size relationships to the definiendum. Firstly we marked the continuous words that describe this relationships with special characters. We marked their relationship with one of the three chosen relationships and we also marked the position. If the definiendum occurred after the genus, we would mark it with a different class than if it occurred before it. In the end we were predicting 6 classes in total. Our training data contained 826 annotated sentences and the test data contained 354. The sentences were firstly tokenized using pretrained model and the classes were encoded into numbers. The sentences were inputted into pretrained BERT model and then the output was fed into a linear layer. We used Adam optimizer and a cross entropy loss function.

3.1 Attention Guided Graph Convolutional Network

For our second method we use Attention Guided Graph Convolutional Network model which is described in paper [8]. The dependency trees generated from sentences convey rich structural information that is proven useful for relationship extraction from text. However they also include many irrelevant information that do not help in our task. To automate solution for this problem, authors use novel attention based GNN architecture, which learns important dependencies through ”soft-pruning” the tree.

The model consists of three main layers as seen on Figure 1. The first one is Attention Guided layer. Here the input is adjacency matrix of dependency graph. Through multi headed attention, the network is transformed into N fully connected graph where numbers in the new adjacency matrices represent weights of the edges. This tries to encode which links are important for relation extraction. Each output is then fed into a set of Densely Connected layers. With this the model captures more structural information from these new larger graphs. Resulting feature vector has rich local and non-local information about the sentence. Lastly all N feature vectors are concatenated and fed into linear combination layer which finally predicts the appropriate relationship type.

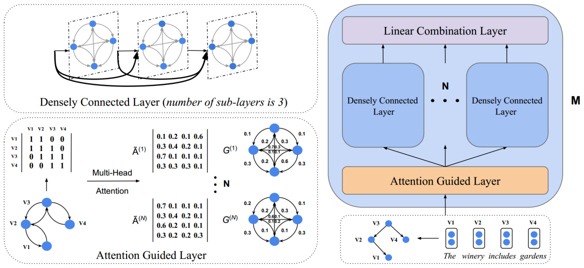

Figure 1. AGGCN model arhitecture [8]

4. Evaluation

4.1 Data

Our primary dataset is TermFrame knowledge base[14] which is a Karstology domain specific corpora for three languages; Slovenian, English and Croatian. In Table 1 we see that the whole corpora is 157 documents in total which were scrapped from various web sources and then hand annotated by linguists.

	English	Slovene	Croatian
Tokens	2,721,042	1,208,240	1,229,368
Words	2,195,982	987,801	969,735
Sentences	97,187	51,990	53,017
Documents	57	60	43

Table 1. The TermFrame corpora information.

TermFrame corpus is annotated with 5 informations; canonical form, semantic category, definition element, semantic relation and relation definitor. Definition element gives us information aout definiendum. Definiendum is the term which is defined in the definition sentence. We will then try to define relationships that are connected to it. There are 16 original relations in total: Genus, Has_size, Has_location, Has_cause, Affects, etc., but we also have to take into account reverse relations where definiendum is after relation. This doubles the total number of classes. Some of them have very few appearances in the corpus, so we won't use them for our model since there likely isn't enough data to learn on.

As our second dataset, to test our models on, we choose SemEval2010 Task-8 dataset as described in paper [2]. It was designed as a benchmark dataset for semantic relation classification in the SemEval competition. The whole corpora contains 10717 sentences with around 200 thousand tokens in total. The goal is to classify noun pairs into one of 11 relations. In Table 2 we see total number and frequency of occurrences for each relation.

Relation name	# Occurances	Frequency
Cause-Effect	1331	12.4%
Component-Whole	1253	11.7%
Entity-Destination	1137	10.6%
Entity-Origin	974	9.1%
Product-Producer	948	8.8%
Member-Collection	923	8.6%
Message-Topic	895	8.4%
Content-Container	732	6.8%
Instrument-Agency	660	6.2%
Other	1864	17,4%

Table 2. SemEval2010 Task 8 dataset semantic relations

4.2 Performance metrics

Our main goal for this task is classifying word pairs into proper relation. Because of that we use standard classification performance metrics to evaluate and compare our models.

Precision is the fraction of relevant instances among the retrieved instances and we calculate it based on Equation 1.

$$Precision = \frac{Tp}{Tp + Fp} \tag{1}$$

Then we calculate recall with Equation 2 which tells us the fraction of relevant instances that were retrieved.

$$Recall = \frac{Tp}{Tp + Fn} \tag{2}$$

And at the end we calculate their harmonic mean with Equation 3 which is called F-score.

$$F - score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{3}$$

Since this is a multi-classification problem and classes are imbalanced we have to calculate measures for each class separately. And because all the classes are equally important we take the unweighted average over all the scores as our final performance.

4.3 Data Preprocessing

We are using models that are already build by the authors of the papers. But before we can use them, we have to transform our data to the shape that they envisioned. In this section we describe our preprocessing pipeline for every used method.

4.3.1 AGGCN Model

This model is based on dependency graphs so we need to generate it for each sentence that we want to use. For this we use Stanza library [15]. Additionally we also tokenize each sentence, perform POS tagging and mark which tokens represent object and subject of the relation. Because the given Karst dataset is not perfect, we have to also define some edge cases when we are matching the right object and subject inside given sentence. After that a vocabulary is created. We download GloVe word embedding[16] and find the most possible amount of matches for our vocabulary. This produces 300 dimensional vector for each one of our matched words. Together with previously acquired dependency graphs, POS tags and locations of object/subjects this represents the input to the model. For Karst dataset we perform stratified 70:30 train-test split so we have same balance over classes in both subsets. For now we choose only ”Genus”, ”Has_Location” and ”Has_size” (and all of them in reverse) relations.

5. Results

To evaluate our models we used precision, recall, F1 score and accuracy. We weighted the calculations according to the frequency of each class in our dataset.

The pretrained DistiledBERT [17] model achieved the accuracy 84.5%, precision 81.1%, recall 84.5% and the F1 score 80.3% on our Karst English train set. On the same data we achieved quite better results with BERT; accuracy 97.3%, precision 97.8%, recall 97.3% and F1 score 97.3%. We also tried RoBERTa [18] that resulted into accuracy 96.2%, precision 96.7%, recall 96.2% and F1 score 96.1%.

5.1 AGGCN Model

We train our model for 150 epochs with SGD optimizer, starting learning rate of 0.5 and 0.9 weight decay every fifth epoch. Architecture uses 3 attention heads and batch size of 50 for SemEval and 32 for Karst dataset.

In Table 3 we can see evaluation results on Karst test subset. We calculate performance for each class separately. We do the same for SemEval dataset and present results in Table 4.

Relation name	Prec	Rec	F-score
Genus	0.736	0.958	0.832
Genus_rev	0.875	0.609	0.718
Has_Location	0.873	0.602	0.713
Has_Location_rev	0.571	0.400	0.451
Has_Size	0.920	0.639	0.754
Has_Size_rev	1	0	0
Overall performance	0.795	0.535	0.64

Table 3. Results of AGGCN model on Karst dataset

Relation name	Prec	Rec	F-score
Cause-Effect	0.918	0.923	0.921
Component-Whole	0.829	0.826	0.828
Content-Container	0.824	0.880	0.851
Entity-Destination	0.865	0.924	0.894
Entity-Origin	0.827	0.856	0.841
Instrument-Agency	0.794	0.769	0.781
Member-Collection	0.835	0.871	0.852
Message-Topic	0.823	0.911	0.865
Product-Producer	0.822	0.861	0.841
Overall performance	0.838	0.870	0.853

Table 4. Results of AGGCN model on SemEval dataset

6. Discussion

TODO: Comment on the results, draw conclusions ...

7. Acknowledgments

References

[1] Evan Sandhaus. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12):e26752, 2008.

[2] Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

[3] Stephen Roller, Douwe Kiela, and Maximilian Nickel. Hearst patterns revisited: Automatic hypernym detection from large text corpora. *CoRR*, abs/1806.03191, 2018.

[4] Maurizio Atzori and Simone Balloccu. Fully-unsupervised embeddings-based hypernym discovery. *Information*, 11(5):268, May 2020.

[5] Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *COLING 1992 Volume 2: The 14th International Conference on Computational Linguistics*, 1992.

[6] Georgeta Bordea, Els Lefever, and Paul Buitelaar. SemEval-2016 task 13: Taxonomy extraction evaluation (TExEval-2). In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1081–1091, San Diego, California, June 2016. Association for Computational Linguistics.

[7] Jose Camacho-Collados, Claudio Delli Bovi, Luis Espinosa-Anke, Sergio Oramas, Tommaso Pasini, Enrico Santus, Vered Shwartz, Roberto Navigli, and Horacio Saggion. SemEval-2018 task 9: Hypernym discovery. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 712–724, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[8] Zhijiang Guo, Yan Zhang, and Wei Lu. Attention guided graph convolutional networks for relation extraction, 03 2022.

[9] Sunil Kumar Sahu, Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. Inter-sentence relation extraction with document-level graph convolutional neural network, 03 2022.

[10] Zhepei Wei, Jianlin Su, Yue Wang, Yuan Tian, and Yi Chang. A novel cascade binary tagging framework for relational triple extraction. *arXiv preprint arXiv:1909.03227*, 2019.

[11] Kang Zhao, Hua Xu, Yue Cheng, Xiaoteng Li, and Kai Gao. Representation iterative fusion based on heterogeneous graph neural network for joint entity and relation extraction. *Knowledge-Based Systems*, 219:106888, 2021.

[12] Amir DN Cohen, Shachar Rosenman, and Yoav Goldberg. Relation classification as two-way span-prediction. *arXiv preprint arXiv:2010.04829*, 2020.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[14] Špela Vintar, Amanda Saksida, Katarina Vrtovec, and Uroš Stepišnik. Modelling specialized knowledge with conceptual frames: The termframe approach to a structured visual domain representation. 2019.

[15] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2020.

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

[17] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108, 2019.

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