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 and English. 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

3.1 BERT

As for the first method we implemented relation extraction with BERT [13]. Firstly we marked the continuous words that describe this relationships with special characters. We marked their relationship with one of the 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. At the end the model had a linear layer predicting 14 classes in total. The sentences were firstly tokenized using pretrained model and the classes were encoded into numbers. The sentences were input into pretrained BERT model and then the output was fed into a linear layer. We used Adam optimizer and a cross entropy loss function. We fine-tuned the pretrained model on the domain dataset and tested the performance. We tried using BERT [13], DistilBERT [14] and RoBERTa [15] and CroSloEngual BERT [16]. The main differences between models are that RoBERTa was trained on 10 times more data than BERT, while DistilBERT uses only half the number of parameters. CroSloEngual BERT was trained only on three languages (Croatian, Slovenian and English) and therefore better than a multilingual BERT model on tasks in this three languages.

We also use BERT model for token classification in order to extract relations from un-annotated sentences. For training our model we input tokenized sentences that are POS tagged with the use of Stanza library[17]. Additionally we mark all tokens that we want to classify with their labels. If we look at for example definiendum we mark start token with ”B-DEFINIENDUM” and all other tokens that are part of this group with ”I-DEFINIENDUM”. This is standard way of data representation for our pretrained model. We try different models and training schemes to obtain the best possible results on English and Slovene data.

3.2 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.

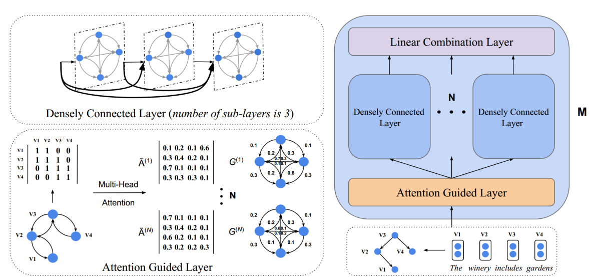

Figure 1. AGGCN model arhitecture [8]

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 [17]. 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[18] 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.

4. Evaluation

4.1 Data

Our primary dataset is TermFrame knowledge base[19] 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 about definiendum. Definiendum is the term which is defined in the definition sentence. We then tried 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 didn't use them in all of our models, since there isn't enough data to learn on. At the end we focused on 7 most frequent relationships (Genus, Has_form, Has_location, Has_cause, Composition_medium, Has_size, and Has_function). We present number and frequency of occurrences for each of these classes in Table 2. Counting every Definiendum-Relation pair separately there are 2641 of them in English part of the corpus and 2706 in Slovene.

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 3 we see total number and frequency of

Relation name	**# Occurances**		**Frequency**	
	EN	SL	EN	SL
Genus	619	643	29.7%	32.2%
Genus_rev	81	112	3.9%	5.6%
Has_form	347	292	16.7%	14.6%
Has_form_rev	53	28	2.5%	1.4%
Has_location	325	311	15.6%	15.6%
Has_location_rev	26	34	1.2%	1.7%
Has_cause	201	208	9.6%	10.4%
Has_cause_rev	22	14	1.1%	0.7%
Comp_medium [1]	143	95	6.9%	4.8%
Comp_medium_rev [2]	30	8	1.4%	0.4%
Has_size	117	103	5.6%	5.2%
Has_size_rev	12	10	0.6%	0.5%
Has_function	95	131	4.6%	6.6%
Has_function_rev	13	11	0.6%	0.6%

Table 2. Karst train dataset semantic relations

occurrences for each relation. We can notice that there are no reverse relations in this dataset.

Relation name	**# Occurrences**	**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 3. 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. We calculate precision, recall and then F-score which is used as a main metric to evaluate how good the model is. 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.

We also need to evaluate relationship extraction. Similarly to word pair classification problem, we can here also use metrics mentioned above. For each word that is tagged as some relation we check if the tag is correct. Based on this precision, recall and F-score are again calculated per tag class and averaged together.

In many cases there are multiple consecutive words that are a part of the same relation in a sentence. Although ideally we would want all of them to be classified correctly, we could

[1] Abbreviation for Composition_medium
[2] Abbreviation for Composition_medium_rev

also allow some room for error, especially for conjunctions or similar types of words. Based on this we also calculate "Soft F-score" which treats a relation to be correctly classified if there is at least 80% matching between true positive and prediction block.

5. Results

5.1 Relation classification

In the first part we tested how good are our models on a task of classifying an object-subject pair into correct relation. We train our AGGCN 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. BERT models were fine-tuned on just 5 epochs. Adam optimizer was used with a learning rate of 2e-05 and batch size of 16.

Firstly, we test our models on SemEval dataset. In Table 4 we see AGGCN performance per class and its overall score. We also test three different versions of BERT models on the same dataset and look at their overall performance in Table 5. From this results we see that RoBERTa worked the best, as it was expected for the biggest model. We decided to use it as it performed the best on a similar task.

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

Model name	Prec	Rec	F-score
BERT	0.808	0.825	0.800
DistilBERT	0.803	0.815	0.793
RoBERTa	**0.817**	**0.823**	**0.803**
CroSloEngual	0.790	0.814	0.782

Table 5. Overall results of different BERT models on SemEval dataset

After that we train and evaluate AGGCN model on English part of TermFrame dataset. Table 6 shows overall results when predicting all available classes (32 of them). We also compare this results to classifications of our BERT model.

When we closely inspect which classes cause the most amount of problems we see the expected results. Some of them are "Affects", "Measures", "Occures in time", etc. This

Model name	Prec	Rec	F-score
AGGCN	0.319	0.272	0.294
RoBERTa	0.263	0.226	0.243

Table 6. Overall results when classifying all TermFrame EN classes

are also the classes that are underrepresented in our training set.

This is why we then train and test classification models just on a subset of most frequently occurring classes as seen in Table 2. In Table 7 we can see per class and overall performance of AGGCN classificator on an EN TermFrame corpus. We can see that the test set doesn't include reverse relations so they are omitted from the results. Genus relations are the easiest to classify which is expected since in many sentences there are "is-a" or other similar patterns that connect definiendum and genus. On the other hand, our model has a lot of trouble with false negative predictions of "Has_function" relation which is shown in very low recall and subsequently low F-score.

Relation name	Prec	Rec	F-score
Composition_medium	0.764	0.500	0.604
Genus	0.834	0.982	0.902
Has_cause	0.655	0.542	0.593
Has_form	0.538	0.636	0.583
Has_function	1	0.071	0.133
Has_location	0.435	0.566	0.492
Has_size	1	0.642	0.782
Overall performance	0.750	0.563	0.642

Table 7. Results of AGGCN model on EN TermFrame (using subset of all classes)

We do the same evaluation by training and testing RoBERTa model on English part of TermFrame dataset and present results in Table 8. Genus and Has_size relations are learned the best while similarly to AGGCN this model also has the most amount of trouble with Has_function relation.

Relation name	Prec	Rec	F-score
Composition_medium	0.857	0.619	0.698
Genus	1	0.991	0.995
Has_cause	0.9	0.633	0.716
Has_form	0.941	0.765	0.829
Has_function	0.357	0.286	0.310
Has_location	0.923	0.615	0.695
Has_size	0.923	0.923	0.923
Overall performance	0.843	0.690	0.759

Table 8. Results of RoBERTa model on EN TermFrame (using subset of all classes)

From our results on SemEval and English part of TermFrame datasets we see that RoBERTa gave us comparable or much better performance compared to AGGCN so this is what we used going forward.

In Table 9 we see results of RoBERTa and CroSloEngual versions of BERT model trained and tested on Slovene corpus. We can see that CroSloEngual majorly outperforms RoBERTa. This is expected as multilingual models perform worse than monolingual or in our case trilingual models.

Relation	**Prec**		**Rec**		**F-score**	
	R	C	R	C	R	C
CM [1]	0.0	0.83	0.0	0.83	0.0	0.83
CM_rev [2]	0.0	1	0.0	1	0.0	1
G	0.93	0.99	0.66	0.91	0.75	0.94
G_rev	0.97	1	0.90	0.97	0.93	0.98
HS	0.56	0.89	0.39	0.72	0.44	0.78
HS_rev	0.0	0.50	0.0	0.25	0.0	0.33
HF	0.97	1	0.62	0.85	0.73	0.91
HF_rev	0.0	0.50	0.0	0.38	0.0	0.42
HFun	0.0	0.67	0.0	0.67	0.0	0.67
HFun_rev	/	/	/	/	/	/
HL	0.87	1	0.74	0.78	0.78	0.86
HL_rev	1	1	0.5	1	0.63	1
HS	0.0	1	0.0	0.92	0.0	0.95
HS_rev	0.0	0.0	0.0	0.0	0.0	0.0
Overall	0.33	**0.75**	0.35	**0.74**	0.32	**0.73**

Table 9. Results of RoBERTa (R) and CroSloEngual (C) on Slovene TermFrame (using subset of all classes)

We also tried using CroSloEngual as a zero shot learning model. We fine-tuned the model on English TermFrame dataset and then tested it on the Slovenian test set. It achieved the overall precision of 0.685, recall of 0.682 and the F1-score of 0.658. We can see that the model performed worse than when it was also fine-tuned on the Slovenian corpus. However, the results were not the worst. This means that the model learns how the relations are formed which can be transferred in both languages.

5.2 Relation extraction

In the second part of our task we focus on extracting relations from an un-annotated sentence and classifying them into the right labels. In our first pipeline we train our model on English part of Karst corpus. As discussed before, some classes are severely underrepresented so we are only focusing on predicting the ones listed in Table2 with addition to ”DEFINIENDUM” class.

In Table 10 we can see results of BERT token classificator evaluated on test dataset. Our model is quite good at labeling definiendum and genus while other classes do not have such good results. One of the reasons for such results for other classes is that this are normally multi-word relations and are thus harder for the model to learn. It seems that the model is not able to capture full meaning for such long representations. The other possible problem is that defininendum and genus are the the most common classes in train and test set by a big margin. The performance would definitely improve if we had more data. This is also confirmed by looking at weighted results where our model gets F-score of 56%.

Additionally we also trained and evaluated models for each relation separately e.g. model that only labels Definiendum and Has_cause or only Definiendum and Has_form. We hypothesized that the results would be better if our model focuses only on one relation at a time, but we saw that performance was only slightly improved and in some cases even decreased. This approach would additionally come with problem of overlapping tags where a token could be classified into more classes. Because of this we abandoned this idea.

Token tag	**Prec**	**Rec**	**F-score**
Composition_medium	0.06	0.06	0.06
Definiendum	0.91	0.95	0.93
Genus	0.44	0.47	0.46
Has_cause	0.06	0.06	0.06
Has_form	0.14	0.38	0.21
Has_function	0.03	0.05	0.04
Has_location	0.06	0.08	0.07
Has_size	0.20	0.31	0.24
Overall performance	0.24	0.29	0.26

Table 10. BERT token classification on EN TermFrame

For our second pipeline we explore merging of token classification and relation classification. Here we first train similar token classificator as before but this time we only use ”DEFINIENDUM” and ”SUBJECT” classes. This means that we tag all relations into the same class but still keep information where each of the relations start with ”B-” and ”I-” tokens. In Table 11 we see the results. As expected definiendum is quite accurately tagged while results of subject tagging are not as good since is difficult for model to encapsulate all picked relations into one class.

Token tag	**Prec**	**Rec**	**F-score**
Definiendum	0.81	0.86	0.83
Subject	0.27	0.28	0.28
Overall performance	0.54	0.57	0.55

Table 11. Token classification into Object/Subject on EN TermFrame

We then use this model to make predictions on the test set and use them as an input to BERT relation classificator. We do this by extracting every object-subject pair and try to predict its relation. Relation classificator has now additional class ”Other” for pairs that are some other relations or not a relation at all.

Similarly as with English, we also train and evaluate BERT Token Classificator on Slovene TermFrame corpus. We see results in Table 12.

Token tag	Prec	Rec	F-score
Composition_medium	0	0	0
Composition_medium_rev	0	0	0
Definiendum	0.78	0.79	0.79
Genus	0.62	0.59	0.61
Genus_rev	0.21	0.15	0.17
Has_cause	0.04	0.06	0.04
Has_cause_rev	0	0	0
Has_form	0.41	0.34	0.37
Has_form_rev	0	0	0
Has_function	0	0	0
Has_function_rev	0	0	0
Has_location	0.31	0.34	0.32
Has_location_rev	0	0	0
Has_size	0.14	0.08	0.11
Has_size_rev	0	0	0
Overall performance	0.17	0.16	0.16

Table 12. BERT token classification on SL TermFrame

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] 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.

[15] 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.

[16] M. Ulčar and M. Robnik-Šikonja. FinEst BERT and CroSloEngual BERT: less is more in multilingual models. In P Sojka, I Kopeček, K Pala, and A Horák, editors, *Text, Speech, and Dialogue TSD 2020*, volume 12284 of *Lecture Notes in Computer Science*. Springer, 2020.

[17] 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 Associa-*

tion for Computational Linguistics: System Demonstrations, 2020.

[18] 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.

[19] 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.