SIRIUS-LTG-UiO at SemEval-2018 Task 7: Convolutional Neural Networks with Shortest Dependency Paths for Semantic Relation Extraction and Classification in Scientific Papers

Farhad Nooralahzadeh, Lilja Øvrelid, Jan Tore Lønning
Department of Informatics
University of Oslo, Norway
{farhadno,liljao,jtl}@ifi.uio.no

Abstract

This article presents the SIRIUS-LTG-UiO system for the SemEval 2018 Task 7 on Semantic Relation Extraction and Classification in Scientific Papers. First we extract the shortest dependency path (sdp) between two entities, then we introduce a convolutional neural network (CNN) which takes the shortest dependency path embeddings as input and performs relation classification with differing objectives for each subtask of the shared task. This approach achieved overall F1 scores of 76.7 and 83.2 for relation classification on clean and noisy data, respectively. Furthermore, for combined relation extraction and classification on clean data, it obtained F1 scores of 37.4 and 33.6 for each phase. Our system ranks 3rd in all three sub-tasks of the shared task.

1 Introduction

Relation extraction and classification can be defined as follows: given a sentence where entities are manually annotated, we aim to identify the pairs of entities that are instances of the semantic relations of interest and classify them based on a pre-defined set of relation types. A range of different approaches have been applied to solve this task in previous work. Conventional classification approaches have made use of contextual, lexical and syntactic features combined with richer linguistic and background knowledge such as WordNet and FrameNet (Hendrickx et al., 2010; Rink and Harabagiu, 2010).

Recently, the re-emergence of deep neural networks provides a way to develop highly automatic features and representations to handle complex interpretation tasks. These approaches have yielded impressive results for many different NLP tasks. The use of deep neural networks for relation classification has been investigated in several recent studies (Socher et al., 2012; Lin et al., 2016; Zhou et al., 2016). Convolutional neural networks (CNNs) have been effectively applied to extract lexical and sentence level features for relation classification (Zhang and Wang, 2015; Lee et al., 2017; Nguyen and Grishman, 2015). However, these works consider whole sentences or the context between two target entities as input for the CNN. Such representations suffer from irrelevant sub-sequences or clauses when target entities occur far from each other or there are other target entities in the same sentence. To avoid negative effects from irrelevant chunks or clauses and capture the relation between two entities, Xu et al. (2015a); Liu et al. (2015) and Xu et al. (2015b) employ a CNN to learn more robust and effective relation representations from the shortest dependency path (sdp) between two entities. The sdp between two entities in the dependency graph captures a condensed representation of the information required to assert a relationship between two entities (Bunescu and Mooney, 2005). In this work, we continue this line of work and present a system based on a CNN architecture over shortest dependency paths combined with domain-specific word embeddings to extract and classify semantic relations in scientific papers.

2 System description

In this section, we describe the various components of our system.

Text pre-processing. For each relation instance in the training data set, we assign a sentence that contains the participant entities. Sentence and token boundaries are detected using the Stanford CoreNLP tool (Manning et al., 2014). Since most of the entities are multi-word units, in order to obtain a precise dependency path between entities, we replace the entities with their codes. The example sentence in (1) below is thus transformed to

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 805–810
New Orleans, Louisiana, June 5–6, 2018. ©2018 Association for Computational Linguistics

(2).

(1) Syntax-based statistical machine translation (MT) aims at applying statistical models to structured data .

(2) P05-1067.1 aims at applying P05-1067.2 to P05-1067.3 .

Given an encoded sentence, we find the sdp connecting two target entities for each relation instance using a syntactic parser, see below.

For syntactic parsing we employ the parser described in Bohnet and Nivre (2012), a transition-based parser which performs joint PoS-tagging and parsing. We train the parser on the standard training sections 02-21 of the Wall Street Journal (WSJ) portion of the Penn Treebank (Marcus et al., 1993). The constituency-based treebank is converted to dependencies using two different conversion tools: (i) the pennconverter software[1] (Johansson and Nugues, 2007), which produces the so-called CoNLL-style dependencies employed in the CoNLL08 shared task on dependency parsing (Surdeanu et al., 2008)[2], and (ii) the Stanford parser using the option to produce basic Stanford dependencies (de Marneffe et al., 2014)[3]. The parser achieves a labeled accuracy score of 91.23 when trained on the CoNLL08 representation and 91.31 for the Stanford basic model, when evaluated against the standard evaluation set (section 23) of the WSJ. We also experimented with the pre-trained parsing model for English included in the Stanford CoreNLP toolkit (Manning et al., 2014), which outputs Universal Dependencies. However, it was clearly outperformed by our version of the Bohnet and Nivre (2012) parser in the initial development experiments.

Based on the dependency graphs output by the parser, we extract the shortest dependency path connecting two entities. The path records the direction of arc traversal using left and right arrows (i.e. ← and →) as well as the dependency relation of the traversed arcs and the predicates involved, following Xu et al. (2015a). The entity codes in the final sdp are replaced with the corresponding word tokens at the end of the pre-processing step. For the sentence in (1) and the two entities *statistical models* and *structured data* we thus extract the path in (3) below.

(3) `statistical models ← OBJ ← applying → DIR → to → PMOD → structured data`

Label encoding. The classification sub-tasks contain five asymmetric relations (USAGE, RESULT, MODEL-FEATURE, PART_WHOLE, TOPIC) and one symmetric relation (COMPARE). The relation instance along with its directionality are provided in both the training and the test data sets. For these sub-tasks we therefore use the same labels in our system. For sub-task 2 which combines the extraction and classification tasks, however, we construct an extra set of relation types. First, we collect every pair of entities within a single sentence that are not included in the annotated relation set. To minimize the noise, we retain only the entity pairs which are not further away than 6 tokens. From these entity pairs we generate negative instances with the NONE class and extract the corresponding sdp. Second, to preserve the directionality in the asymmetric relations, we add the ¬ symbol to the instances with reverse directionality (e.g., USAGE(e1,e2,REVERSE) becomes ¬USAGE(e1,e2)). The final label set for sub-task 2 thus consists of 12 relations.

Word embeddings. In our system, two different sets of pre-trained word embeddings are used for initialization. One is the 300-d pre-trained embeddings provided by the NLPL repository [4](Fares et al., 2017), trained on English Wikipedia data with word2vec (Mikolov et al., 2013), here dubbed wiki-w2v. In addition, we train a second set of domain-specific embeddings on the ACL Anthology corpus. We obtain the XML versions of 22,878 articles from ACL Anthology [5]. After extracting the raw texts, for training of the 300-d word embeddings (acl-w2v), we exploit the available word2vec (Mikolov et al., 2013) implementation *gensim* (Řehůřek and Sojka, 2010) for training.

Classification Model Our system is based on a Convolutional Neural Network (CNN) architecture similar to the one used for sentence classification in Kim (2014). Figure 1 provides an overview

[1] `http://nlp.cs.lth.se/software/treebank-converter/`

[2] The pennconverter tool is run using the `rightBranching=false` flag.

[3] The Stanford parser is run using the `-basic` flag to produce the basic version of Stanford dependencies.

[4] `http://vectors.nlpl.eu/repository/`

[5] `https://acl-arc.comp.nus.edu.sg/`

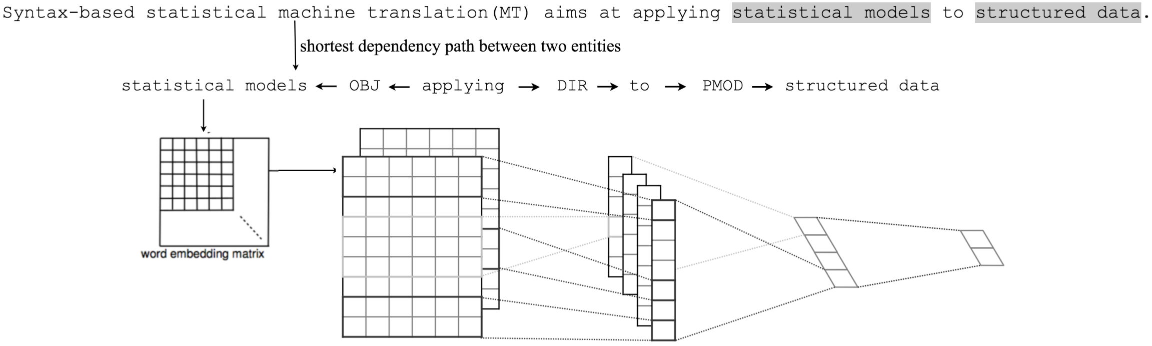
Syntax-based statistical machine translation(MT) aims at applying statistical models to structured data.
shortest dependency path between two entities
statistical models ← OBJ ← applying → DIR → to → PMOD → structured data
word embedding matrix

Figure 1: Model architecture with two channels for an example shortest dependency path (CNN model from Kim (2014)).

of the proposed model. It consists of 4 main layers as follows:

Look-up Table and Embedding layer: In the first step, the model takes a dependency path, as in (3) as input and transforms it into a matrix representation by looking up the pre-trained word embeddings.

Convolutional Layer: The next layer performs convolutions with the ReLU activation to the embedding layer using multiple filter sizes (*filter_sizes* $\in [3, 4, 5]$) and extracts feature maps over the tokens.

Max pooling Layer: By applying the *max* operator, the most effective local features are generated from each feature map.

Fully connected Layer: Finally, the higher level syntactic features are fed to a fully connected *softmax* layer which outputs the probability distribution over each relation.

3 Experiments

Dataset For each sub-task, the training data includes abstracts of papers from the ACL Anthology corpus with pre-annotated entities. For subtask 1.1 and 2, the training datasets are the same. It contains entities that are manually annotated and they represent domain concepts specific to Natural Language Processing (NLP). In sub-task 1.2 the entities are automatically assigned and therefore contain a fair amount of noise (verbs, irrelevant words). The terms include high-level terms (e.g. ”algorithm”, ”paper”, ”method”) and are not always full NPs (Gábor et al., 2018). Since the related entity pairs and the relation types are provided for the full dataset, we extend the dataset for sub-task 1.1 and 2 by extracting the related entities and their corresponding sdp from the sub-task 1.2 dataset. In order to train a model for sub-task 2, we also augment the dataset by extracting NONE relation instances (see Section 2), extracted from the corresponding dataset. Table 1 shows the number of instances for each relation class. As we can see, the class distribution is clearly unbalanced.

	Subtask		**Reverse**		
Relation	1.1 & 2	1.2	False	True	**Total**
USAGE	483	464	615	332	947
MODEL-FEATURE	326	172	346	152	498
RESULT	72	121	135	58	193
TOPIC	18	240	235	23	258
PART_WHOLE	233	192	273	152	425
COMPARE	95	41	136	-	136
NONE	2315	-	2315	-	2315

Table 1: Number of instances for each relation in the final dataset.

Model settings We keep the value of hyperparameters equal to the ones that are reported in the original work (Kim, 2014), i.e., 128 filters for each window size, a dropout rate of $\rho = 0.5$ and l_2 regularization of 3. To deal with the effects of class imbalance, we weight the cost by the ratio of class instances, thus each observation receives a weight, depending on the class it belongs to. The effect of the minority class observations is thereby increased simply by a higher weight of these instances and is decreased for majority class observations. Furthermore, to guarantee that each fold in n-fold cross validation will have the proportion of same classes during training, evaluation and test, we apply the stratification technique proposed by Sechidis et al. (2011). We use the validation set to detect when overfitting starts during the training of our model; using *early stopping*, training is

			F1	
Sub-task	**Model**	**Representation**	Ext.	Class.
1.1	cnn.multi.acl-w2v.rand	Stanford Basic	-	74.16
1.2			-	77.70
2	cnn.acl-w2v	CoNLL08	74.26	60.31

Table 2: F1 (macro-average) scores for selected configurations during training.

then stopped before convergence to avoid overfitting (Prechelt, 1998). The official evaluation metric is the macro-averaged F1-score, therefore we implement early-stopping (*patience*= 20) based on macro-F1 score in the development set.

Model variants We run experiments with several variants of the model as follows: `cnn.rand`: A baseline model, where all elements in the embedding layer are randomly initialized and updated in the training process. `cnn.wiki-w2v`: The embedding layer is initialized with the pre-trained Wikipeida word embeddings and fine-tuned for the target task. `cnn.acl-w2v`: The embedding layer is initialized with the pre-trained ACL Anthology word embeddings and fine-tuned for the target task. `cnn.multi.rand`: There are two embedding layers as a 'channel' in the CNN architecture. Both channels are initialized randomly and only one of them is updated during training while the other remains static. `cnn.multi.wiki-w2v`: Same as before, but the channels are initialized with Wikipedia embedding vectors. `cnn.multi.acl-w2v`: The two channels are initialized with ACL embedding vectors. `cnn.multi.wiki-w2v.rand`: First channel is initialized with Wikipedia embeddings in static mode and the second initialized randomly with a non-static mode. `cnn.multi.acl-w2v.rand`: Same as previous setting, but the first channel makes use of ACL embeddings.

Results During development, we investigate the performance of different configurations; different dependency representations (CoNLL08 and Stanford basic) and model variants (see above); by running 5-fold cross validation (i.e. 3 folds for training, 1 fold for evaluation and 1 fold for test). The experiments show that, the multi-channel mode performs better only in the classification sub-tasks compared to the single channel setting. The results suggest that having a significant amount of instances per relation assists the model to classify better. The use of the pre-trained embeddings helps the model in class assignment. Particularly, the domain-specific embeddings (i.e. acl-w2v) provide higher performance gains when used in the model. Table 2 presents the F1-score of the best performing model for each sub-task via 5-fold cross validation on the training data. In the evaluation period, we re-run 5-fold cross validation using selected model for each sub-task. However, in this setting we use 4 folds as training and 1 fold as development set, and we apply the output model to the evaluation dataset. We select the 1st and 2nd best performing models on the development datasets as well as the majority vote (mv) of 5 models for the final submission. The final results are shown in Table 3.

	1st		**2nd**		***mv***	
Sub-task	Ext.	Class.	Ext.	Class.	Ext.	Class.
1.1	-	72.1	-	74.7	-	**76.7**
1.2	-	**83.2**	-	82.9	-	80.1
2	**37.4**	**33.6**	36.5	28.8	35.6	28.3

Table 3: Official evaluation results of the submitted runs on the test set.

4 Conclusion

We present a CNN model over shortest dependency paths between entity pairs for relation extraction and classification. We examine various architectures for the proposed model. The experiments demonstrate the effectiveness of domain-specific word embeddings for all sub-tasks as well as sensitivity to the specific dependency representation employed in the input layer. Our future work includes: 1) to perform error analysis for the different sub-tasks, and 2) to investigate the effects of different dependency representations in relation extraction and classification.

References

Bernd Bohnet and Joakim Nivre. 2012. A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. In *Proceedings of EMNLP*, pages 1455–1465, Jeju Island, Korea. ACL.

Razvan C. Bunescu and Raymond J. Mooney. 2005. 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, pages 724–731, Stroudsburg, PA, USA. Association for Computational Linguistics.

Murhaf Fares, Andrey Kutuzov, Stephan Oepen, and Erik Velldal. 2017. Word vectors, reuse, and replicability: Towards a community repository of large-text resources. In *Proceedings of the 21st Nordic Conference on Computational Linguistics, NoDaLiDa, 22-24 May 2017, Gothenburg, Sweden*, 131, pages 271–276. Linköping University Electronic Press, Linköpings universitet.

Kata Gábor, Davide Buscaldi, Anne-Kathrin Schumann, Behrang QasemiZadeh, Haïfa Zargayouna, and Thierry Charnois. 2018. Semeval-2018 Task 7: Semantic relation extraction and classification in scientific papers. In *Proceedings of International Workshop on Semantic Evaluation (SemEval-2018)*, New Orleans, LA, USA.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó. Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. 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. Association for Computational Linguistics.

Richard Johansson and Pierre Nugues. 2007. Extended constituent-to-dependency conversion for english. In *NODALIDA 2007 Proceedings*, pages 105–112. University of Tartu.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751. Association for Computational Linguistics.

Ji Young Lee, Franck Dernoncourt, and Peter Szolovits. 2017. MIT at semeval-2017 task 10: Relation extraction with convolutional neural networks. *CoRR*, abs/1704.01523.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2124–2133. Association for Computational Linguistics.

Yang Liu, Furu Wei, Sujian Li, Heng Ji, Ming Zhou, and Houfeng Wang. 2015. A dependency-based neural network for relation classification. *CoRR*, abs/1507.04646.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpora of English. The Penn Treebank. *Journal of Computational Linguistics*, 19:313–330.

Marie-Catherine de Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. 2014. Universal Stanford dependencies. A cross-linguistic typology. In *International Conference on Language Resources and Evaluation*, pages 4585–4592, Reykjavik, Iceland.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Thien Huu Nguyen and Ralph Grishman. 2015. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 39–48. Association for Computational Linguistics.

Lutz Prechelt. 1998. Early stopping-but when? In *Neural Networks: Tricks of the Trade, This Book is an Outgrowth of a 1996 NIPS Workshop*, pages 55–69, London, UK, UK. Springer-Verlag.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA.

Bryan Rink and Sanda Harabagiu. 2010. Utd: Classifying semantic relations by combining lexical and semantic resources. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval '10, pages 256–259, Stroudsburg, PA, USA. Association for Computational Linguistics.

Konstantinos Sechidis, Grigorios Tsoumakas, and Ioannis Vlahavas. 2011. On the stratification of multi-label data. In *Proceedings of the 2011 European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part III*, ECML PKDD'11, pages 145–158, Berlin, Heidelberg. Springer-Verlag.

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In

Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 1201–1211. Association for Computational Linguistics.

Mihai Surdeanu, Richard Johansson, Adam Meyers, Lluís Màrquez, and Joakim Nivre. 2008. The CoNLL 2008 Shared Task on Joint Parsing of Syntactic and Semantic Dependencies. In *Proceedings of the Conference on Natural Language Learning*, pages 159–177, Manchester, UK.

Kun Xu, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2015a. Semantic relation classification via convolutional neural networks with simple negative sampling. *CoRR*, abs/1506.07650.

Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015b. Classifying relations via long short term memory networks along shortest dependency path. *CoRR*, abs/1508.03720.

Dongxu Zhang and Dong Wang. 2015. Relation classification via recurrent neural network. *CoRR*, abs/1508.01006.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.