GU IRLAB at SemEval-2018 Task 7: Tree-LSTMs for Scientific Relation Classification

Sean MacAvaney, Luca Soldaini, Arman Cohan, and Nazli Goharian
Information Retrieval Lab
Department of Computer Science
Georgetown University
{firstname}@ir.cs.georgetown.edu

Abstract

SemEval 2018 Task 7 focuses on relation extraction and classification in scientific literature. In this work, we present our tree-based LSTM network for this shared task. Our approach placed 9th (of 28) for subtask 1.1 (relation classification), and 5th (of 20) for subtask 1.2 (relation classification with noisy entities). We also provide an ablation study of features included as input to the network.

1 Introduction

Information Extraction (IE) has applications in a variety of domains, including in scientific literature. Extracted entities and relations from scientific articles could be used for a variety of tasks, including abstractive summarization, identification of articles that make similar or contrastive claims, and filtering based on article topics. While ontological resources can be leveraged for entity extraction (Gábor et al., 2016), relation extraction and classification still remains a challenging task. Relations are particularly valuable because (unlike simple entity occurrences) relations between entities capture lexical semantics. SemEval 2018 Task 7 (Semantic Relation Extraction and Classification in Scientific Papers) encourages research in relation extraction in scientific literature by providing common training and evaluation datasets (Gábor et al., 2018). In this work, we describe our approach using a tree-structured recursive neural network, and provide an analysis of its performance.

There has been considerable previous work with scientific literature due to its availability and interest to the research community. A previous shared task (SemEval 2017 Task 10) investigated the extraction of both keyphrases (entities) and relations in scientific literature (Augenstein et al., 2017). However, the relation set for this shared task was limited to just synonym and hypernym relationships. The top three approaches used for relation-only extraction included convolutional neural networks (Lee et al., 2017a), bi-directional recurrent neural networks with Long Short-Term Memory (LSTM, Hochreiter and Schmidhuber, 1997) cells (Ammar et al., 2017), and conditional random fields (Lee et al., 2017b).

There are several challenges related to scientific relation extraction. One is the extraction of the entities themselves. Luan et al. (2017) produce the best published results on the 2017 ScienceIE shared task for entity extraction using a semi-supervised approach with a bidirectional LSTM and a CRF tagger. Zheng et al. (2014) provide an unsupervised technique for entity linking scientific entities in the biomedical domain to an ontology.

Contribution. Our approach employs a tree-based LSTM network using a variety of syntactic features to perform relation label classification. We rank 9th (of 28) when manual entities are used for training, and 5th (of 20) when noisy entities are used for training. Furthermore, we provide an ablation analysis of the features used by our model. Code for our model and experiments is available.[1]

2 Methodology

Syntactic information between entities plays an important role in relation extraction and classification (Mintz et al., 2009; MacAvaney et al., 2017). Similarly, sequential neural models, such as LSTM, have shown promising results on scientific literature (Ammar et al., 2017). Therefore, in our approach, we leverage both syntactic structures and neural sequential models by employing a tree-based long-short term memory cell (tree-LSTM). Tree-LSTMs, originally introduced by Tai et al. (2015), have been successfully used to

[1] https://github.com/Georgetown-IR-Lab/semeval2018-task7

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 831–835
New Orleans, Louisiana, June 5–6, 2018. ©2018 Association for Computational Linguistics

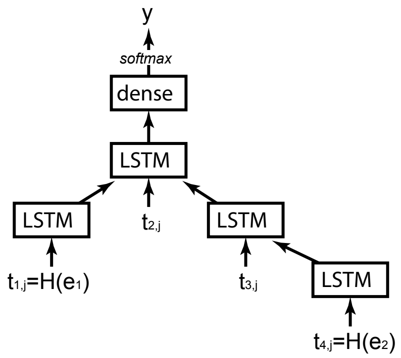


Figure 1: Our tree LSTM network.

Relation (abbr.)	Example
USAGE (U)	Oral communication may offer additional indices...
MODEL-FEATURE (M-F)	We look at the intelligibility of MT output...
PART_WHOLE (P-W)	As the operational semantics of natural language applications improve...
COMPARE (C)	Bag-of-words methods are shown to be equivalent to segment order-sensitive methods in terms of...
RESULT (R)	We find that interpolation methods improve the performance...
TOPIC (T)	A formal analysis for a large class of words called alternative markers...

Table 1: Example relations for each type. Entities are underlined, and all relations are from the first entity to the second entity (non-reversed).

capture relation information in other domains (Xu et al., 2015; Miwa and Bansal, 2016). On a high level, tree-LSTMs operate very similarly to sequential models; however, rather than processing tokens sequentially, they follow syntactic dependencies; once the model reaches the root of the tree, the output is used to compute a prediction, usually through a dense layer. We use the child-sum variant of tree-LSTM (Tai et al., 2015).

Formally, let $S_j = \{t_{1,j}, \ldots, t_{n,j}\}$ be a sentence of length n, $e_1 = \{t_i, \ldots, t_k\}$ and $e_2 = \{t_p, \ldots, t_q\}$ two entities whose relationship we intend to classify; let $\mathrm{H}(e_1)$, $\mathrm{H}(e_2)$ be the root of the syntactic subtree spanning over entities e_1 and e_2. Finally, let $\mathrm{T}(e_1, e_2)$ be the syntactic sub-tree spanning from $\mathrm{H}(e_1)$ to $\mathrm{H}(e_2)$. For the first example in Table 1, $e_1 = \{$*'Oral', 'communication'*$\}$, $e_2 = \{$*'indices'*$\}$, $\mathrm{H}(e_1) = \{$*'communication'*$\}$, $\mathrm{T}(e_1, e_2) = \{$*'communication', 'offer', 'indices'*$\}$. The proposed model uses word embeddings of terms in $\mathrm{T}(e_1, e_2)$ as inputs; the output of the tree-LSTM cell on the root of the syntactic tree is used to predict one of the six relation types (y) using a softmax layer. A diagram of our tree LSTM network is shown in Figure 1.

In order to overcome the limitation imposed by the small amount of training data available for this task, we modify the general architecture proposed in (Miwa and Bansal, 2016) in two crucial ways. First, rather than using the representation of entities as input, we only consider the syntactic head of each entity. This approach improves the generalizability of the model, as it prevents overfitting on very specific entities in the corpus. For example, by reducing *'Bag-of-words methods'* to *'methods'* and *'segment order-sensitive models'* to *'models'*, the model is able to recognize the COMPARE relation between these two entities (see Table 1). Second, we experimented with augmenting each term representation with the following features:

- Dependency labels (DEP): we append to each term embedding the label representing the dependency between the term and its parent.
- PoS tags (POS): the part-of-speech tag for each term is append to its embedding.
- Entity length (ENTLEN): we concatenate the number of tokens in e_1 and e_2 to embeddings representation of heads $\mathrm{H}(e_1)$ to $\mathrm{H}(e_2)$. For terms that are not entity heads, the entity length feature is replaced by '0'.
- Height: the height of each term in the syntactic subtree connecting two entities.

3 Experimental Setup

SemEval 2018 Task 7 focuses on relation extraction, assuming a gold set of entities. This allows participants to focus on specific issues related to relation extraction with a rich set of semantic relations. These include relations for USAGE, MODEL-FEATURE, PART_WHOLE, COMPARE, RESULT, and TOPIC. Examples of each type of relation are given in Table 1.

The shared task evaluates three separate subtasks (1.1, 1.2, and 2). We tuned and submitted

Dataset	U	M-F	P-W	C	R	T
Subtask 1.1						
Train	409	289	215	86	57	15
Valid.	74	37	19	9	15	3
Test	175	66	70	21	20	3
Subtask 1.2						
Train	363	124	162	29	94	207
Valid.	107	51	34	12	29	36
Test	123	75	56	3	29	69

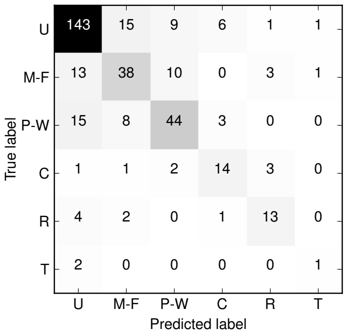
Table 2: Frequency of relation labels in train, validation, and test sets. See Table 1 for relation label abbreviations. Subtask 1.1 uses manual entity labels, and subtask 1.2 uses automatic entity labels (which may be noisy).

System	F1	Rank
Subtask 1.1 (28 teams)		
Our submission	60.9	9
Median team	45.5	
Mean team	37.1	
Subtask 1.2 (20 teams)		
Our submission	78.9	5
Median team	70.3	
Mean team	54.0	

Table 3: Performance result comparison to other task participants for subtasks 1.1 and 1.2.

our system for subtasks 1.1 and 1.2. In both of these subtasks, participants are given scientific abstracts with entities and candidate relation pairs, and are asked to determine the relation label of each pair. For subtask 1.1, both the entities and relations are manually annotated. For subtask 1.2, the entities are automatically generated using the procedure described in Gábor et al. (2016). This procedure introduces noise, but represents a more realistic evaluation environment than subtask 1.1. In both cases, relations and gold labels are produced by human annotators. All abstracts are from the ACL Anthology Reference Corpus (Bird et al., 2008). We randomly select 50 texts from the training datasets for validation of our system. We provide a summary of the datasets for training, validation, and testing in Table 2. Notice how the proportions of each relation label vary considerably among the datasets.

We experiment with two sets of word embeddings: Wiki News and arXiv. The Wiki News embeddings benefit from the large amount of general language, and the arXiv embeddings capture specialized domain language. The Wiki News embeddings are pretrained using fastText with a dimension of 300 (Mikolov et al., 2018). The arXiv embeddings are trained on a corpus of text from the cs section of arXiv.org[2] using a window of 8 (to capture adequate term context) and a dimension of 100 (Cohan et al., 2018). A third variation of the embeddings simply concatenates the Wiki News and arXiv embeddings, yielding a dimension of 400; for words that appear in only one of the two embedding sources, the available embeddings are concatenated with a vector of appropriate size sampled from $\mathcal{N}(0, 10^{-8})$.



True label \ Predicted label	U	M-F	P-W	C	R	T
U	143	15	9	6	1	1
M-F	13	38	10	0	3	1
P-W	15	8	44	3	0	0
C	1	1	2	14	3	0
R	4	2	0	1	13	0
T	2	0	0	0	0	1

Figure 2: Confusion matrix for subtask 1.1.

For our official SemEval submission, we train our model using the concatenated embeddings and one-hot encoded dependency label features. We use a hidden layer of 200 nodes, a 0.2 dropout rate, and a training batch size of 16. Syntactic trees were extracted using SpaCy[3], and the neural model was implemented using MxNet[4].

The official evaluation metric is the macro-averaged F1 score of all relation labels. For additional analysis, we use the macro precision and recall, and the F1 score for each relation label.

4 Results and Discussion

In Table 3, we provide our official SemEval results in the context of other task participants. In both subtasks, we ranked above both the median and mean team scores, treating the top-ranking approach for each team as the team's score. For Subtask 1.1, we ranked 9 out of 28, and for Subtask 1.2, we ranked 5 out of 20. This indicates

[2] https://github.com/acohan/long-summarization

[3] https://spacy.io/

[4] https://mxnet.incubator.apache.org/

	Overall			F1 by label					
Features	P	R	F1	U	M-F	P-W	C	R	T
Subtask 1.1									
(no features)	56.9	64.1	59.5	**81.4**	51.5	59.9	57.8	61.9	44.4
DEP	53.5	54.1	53.6	79.1	55.5	58.2	63.8	**64.9**	0.0
DEP + POS	**60.1**	59.1	59.5	79.9	57.1	58.5	**68.3**	60.0	33.3
DEP + POS + EntLen	59.4	**64.1**	**60.9**	80.0	**59.0**	56.9	58.3	61.1	**50.0**
DEP + POS + EntLen + Height	52.1	53.3	52.4	79.2	57.4	**62.2**	56.0	59.5	0.0
Subtask 1.2									
(no features)	74.2	78.9	75.4	80.0	65.6	72.6	57.1	80.0	**97.1**
DEP	76.4	78.5	76.4	79.2	67.2	73.0	**66.7**	79.4	93.1
DEP + POS	75.5	**80.3**	77.3	**82.0**	**73.9**	**73.6**	57.1	80.0	**97.1**
DEP + POS + EntLen	**78.2**	79.7	**78.0**	81.9	69.3	70.5	**66.7**	**82.5**	**97.1**
DEP + POS + EntLen + Height	73.0	78.7	74.8	79.5	70.7	70.3	57.1	74.3	**97.1**

Table 4: Feature ablation results for subtasks 1.1 and 1.2. DEP are dependency labels, POS are part of speech labels, EntLen is is the length of the input entities, and Height is the height of the entities in the dependency tree. In both subtasks 1.1 and 1.2, the combination of dependency labels, parts of speech, and entity lengths yield the best performance in terms of overall F1 score.

Embeddings	P	R	F1
Subtask 1.1			
Wiki News	59.2	57.3	57.6
arXiv	58.5	55.1	56.4
Wiki News + arXiv	**59.4**	**64.1**	**60.9**
Subtask 1.2			
Wiki News	73.1	76.2	72.7
arXiv	65.4	67.4	65.9
Wiki News + arXiv	**78.2**	**79.7**	**78.0**

Table 5: Performance comparison for subtasks 1.1 and 1.2 when using Wiki News and arXiv embeddings. The concatenated embeddings outperform the individual methods.

that our approach is generally more tolerant to the noisy entities given in Subtask 1.2 than most other approaches. Figure 2 is a confusion matrix for the official submission for subtask 1.1. The three most frequent labels in the training data (USAGE, MODEL-FEATURE, and PART_WHOLE) are also the most frequently confused relation labels. This behavior can be partially attributed to the class imbalance.

In Table 4, we examine the effects of various feature combinations on the model. Specifically, we check the macro averaged precision, recall, and F1 scores for both subtask 1.1 and 1.2 with various sets of features on the test set. Of the combinations we investigated, including the dependency labels, part of speech tags, and the token length of entities yielded the best results in terms of overall F1 score for both subtasks. The results by individual relation label are more mixed, with the overall best combination simply yielding better performance on average, not on each label individually. Interestingly, the entity height feature reduces performance, perhaps indicating that it is easy to overfit the model using this feature.

Table 5 examines the effect of the choice of word embeddings on performance. In both subtasks, concatenating the Wiki News and arXiv embeddings yields better performance than using a single type of embedding. This suggests that the two types of embeddings are useful in different cases; perhaps Wiki News better captures the general language linking the entities, whereas the arXiv embeddings capture the specialized language of the entities themselves.

5 Conclusion

In this work, we investigated the use of a tree LSTM-based approach for relation classification in scientific literature. Our results at SemEval 2018 were encouraging, placing 9th (of 28) at subtask 1.1 (relation classification with manually-annotated entities), and 5th (of 20) at subtask 1.2 (relation classification using automatically-generated entities). Furthermore, we conducted an analysis of our system by varying the system parameters and features.

References

Waleed Ammar, Matthew E. Peters, Chandra Bhagavatula, and Russell Power. 2017. The AI2 system at SemEval-2017 Task 10 (ScienceIE): semi-supervised end-to-end entity and relation extraction. In *SemEval-2017*.

Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew D McCallum. 2017. SemEval 2017 Task 10: ScienceIE - extracting keyphrases and relations from scientific publications. In *SemEval-2017*.

Steven Bird, Robert Dale, Bonnie J. Dorr, Bryan R. Gibson, Mark Thomas Joseph, Min-Yen Kan, Dongwon Lee, Brett Powley, Dragomir R. Radev, and Yee Fan Tan. 2008. The ACL anthology reference corpus: A reference dataset for bibliographic research in computational linguistics. In *LREC*.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Kim Seokhwan Bui, Trung, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *NAACL-HLT*.

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 *SemEval-2018*.

Kata Gábor, Haïfa Zargayouna, Davide Buscaldi, Isabelle Tellier, and Thierry Charnois. 2016. Semantic annotation of the ACL anthology corpus for the automatic analysis of scientific literature. In *LREC*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Ji Young Lee, Franck Dernoncourt, and Peter Szolovits. 2017a. MIT at SemEval-2017 Task 10: Relation extraction with convolutional neural networks. In *SemEval-2017*.

Lung-Hao Lee, Kuei-Ching Lee, and Y Jane Tseng. 2017b. The NTNU system at SemEval-2017 Task 10: Extracting keyphrases and relations from scientific publications using multiple conditional random fields. In *SemEval-2017*.

Yi Luan, Mari Ostendorf, and Hannaneh Hajishirzi. 2017. Scientific information extraction with semi-supervised neural tagging. In *EMNLP*.

Sean MacAvaney, Arman Cohan, and Nazli Goharian. 2017. GUIR at SemEval-2017 Task 12: A framework for cross-domain clinical temporal information extraction. In *SemEval-2017*.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *LREC*.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *ACL/IJCNLP*. Association for Computational Linguistics.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using lstms on sequences and tree structures. In *ACL*.

Kai Sheng Tai, Richard Socher, and Christopher D Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*.

Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015. Classifying relations via long short term memory networks along shortest dependency paths. In *EMNLP*.

Jinguang Zheng, Daniel P Howsmon, Boliang Zhang, Juergen Hahn, Deborah L. McGuinness, James A. Hendler, and Heng Ji. 2014. Entity linking for biomedical literature. In *DTMBIO@CIKM*.