Semantic Frame Labeling with Target-based Neural Model

Yukun Feng[1], **Dong Yu**[1]*, **Jian Xu**[2], **ChunHua Liu**[1]
[1] Beijing Language and Culture University,
[2] University of Science and Technology of China
yukunfg@gmail.com, yudong@blcu.edu.cn,
jianxu1@mail.ustc.edu.cn, chunhualiu596@gmail.com

Abstract

This paper explores the automatic learning of distributed representations of the target's context for semantic frame labeling with target-based neural model. We constrain the whole sentence as the model's input without feature extraction from the sentence. This is different from many previous works in which local feature extraction of the targets is widely used. This constraint makes the task harder, especially with long sentences, but also makes our model easily applicable to a range of resources and other similar tasks. We evaluate our model on several resources and get the state-of-the-art result on subtask 2 of SemEval 2015 task 15. Finally, we extend the task to word-sense disambiguation task and we also achieve a strong result in comparison to state-of-the-art work.

1 Introduction and Related Work

Semantic frame labeling is the task of selecting the correct frame for a given target based on its semantic scene. A target is often called lexical unit which evokes the corresponding semantic frame. The lexical unit can be a verb, adjective or noun. Generally, a semantic frame describes how the lexical unit is used and specifies its characteristic interactions. There are many semantic frame resources, such as FrameNet (Baker et al., 1998), VerbNet (Schuler, 2006), PropBank (Palmer et al., 2005) and Corpus Pattern Analysis (CPA) frames (Hanks, 2012). However, most existing frame resources are manually created, which is time-consuming and expensive. Automatic semantic frame labeling can lead to the development of a broader range of resources.

*The corresponding author

Early works for semantic frame labeling mainly focus on FrameNet, PropBank and VerbNet resources. But most of them focus only one resource and rely heavily on feature engineering (e.g., Honnibal and Hawker 2005; Abend et al. 2008). Recently, there are some works on learning CPA frames based on a new semantic frame resource, the Pattern Dictionary of English Verbs (PDEV) (El Maarouf and Baisa, 2013; El Maarouf et al., 2014). The above two works also rely on features and both are only tested on 25 verbs. Most works aim at constructing the context representations of the target with explicit rules based on some basic features, e.g., Parts Of Speech (POS), Named Entities (NE) and dependency relations related to the target. Currently, some deep learning models have been applied with dependency features. Hermann et al. (2014) used the direct dependents and dependency path to extract the context representation based on distributed word embeddings on English FrameNet. Inspired by the work, Zhao et al. (2016) used a deep feed forward neural network on Chinese FrameNet with similar features. This is different from our goal where we want to explore an appropriate deep learning architecture without complex rules to construct the context representations. Feng et al. (2016) used a multilayer perceptrons (MLP) model on CPA frames without extra feature extraction, but the model is quite simple and has an input window which is not convenient.

In this paper, we present a target-based neural model which takes the whole target-specific sentence as input and gives the semantic frame label as output. Our goal is to make the model light without explicit rules to construct context representations and applicable to a range of resources. To cope with variable-length sentences under our constraint, a simple idea is to use recurrent neural networks (RNN) to process the sentences. But

*Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*, pages 91–96,
Vancouver, Canada, August 3-4, 2017.

noise caused by irrelevant words in long sentences may hinder learning. In fact, the arguments related to the target are usually distributed near the target because when we write or speak, we will focus mainly on arguments that are in the immediate context of a core word. We use two RNNs each of which processes one part of the sentence split by the target. The model takes the target as the center and we call it the target-based recurrent networks (TRNN). In fact, TRNN itself is not novel enough, but according to our knowledge, no related research has focused on this topic. We will show that TRNN is quite suitable for learning the context of the target.

2 Model Description

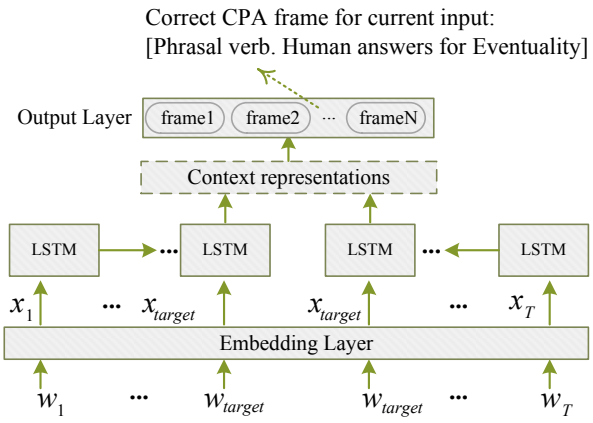

Figure 1: Architecture of TRNN with an example sentence whose target word is in bold.

In our model we select long short-term memory (LSTM) networks, a type of RNN designed to avoid the vanishing and exploding gradients. The overall structure is illustrated Figure 1. w_t is the t-th word in the sentence the length of which is T and $target$ is the index of the target. x_t is obtained by mapping w_t into a fixed vector through well pre-trained word vectors. The model has two LSTMs each of which processes one part of the sentence split by the target. The model can automatically learn the distributed representation of target's context from w with few manual design.

2.1 Context Representations

An introduction about LSTM can be found in the work of Hochreiter and Schmidhuber (1997). The parameters of LSTM are W_{x*}, W_{h*} and b_* where $*$ stands for one of several internal gates. W_{x*} is the matrix between the input vector x_t and gates, W_{h*} is the matrix between the output h_t of LSTM and gates and b_* is the bias vector on gates. The formulas of LSTM are:

$$
\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \\
c_t &= f_t \odot c_{t-1} + i_t \odot \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned}
$$

where σ is the sigmoid function and $\odot$ represents the element-wise multiplication. i_t, f_t c_t and o_t are the output of input gates, forget gates, cell states and output gates, respectively. In our model, two LSTMs share the same parameters. At last, the target's context representations cr are added by the outputs of two LSTMs:

$$cr = h_{target-1} + h_{target}$$

The dimension of cr is decided by the number of hidden units in LSTM, which is a hyper parameter in our model, and is usually much lower than that of one word vector. Here we make some intuitions behind the above formulas. The gradients from last layer flow equally on the $(target-1)$-th LSTM box and the $target$-th LSTM box and then the two flows go to both ends. As it is quite common in deep learning models, the gradients usually become ineffective as the depth of the flow increases especially when the sentence is very long. The gradients on words far from the target get less impact than those near the target. As a whole, more data are usually required to learn the arguments far from the target than those near the target. If the real arguments are distributed near the target, this model will be suitable as its architecture is designed to take care of the local context of the target.

2.2 Output Layer

We use Softmax layer as the output layer on the context representations. The output layer computes a probability distribution over the semantic frame labels. During the training, the cost we minimize is the negative log likelihood of the model:

$$L = -\sum_{m=1}^{M} \log p_{t_m}$$

Here M is number of the training sentences, t_m is the index of the correct frame label for the m-th sentence and p is the probability.

3 Experiments

3.1 Datasets

We simply divide all the datasets in two types: **per-target** and **non per-target**. Per-target semantic frame resources define a different set of frame labels for each target and we train one model for each target; different targets may share some semantic frame labels in non per-target resources and we train a single model for such resources. We use the Semlink project (Loper et al., 2007) to create our datasets [1]. Semlink aims to link together different lexical resources via a set of mappings. We use its corpus which annotates FrameNet and Propbank frames for the WSJ section of the Penn Treebank. Another resource we use is PDEV [2] which is quite new and has CPA frame annotated examples on British National Corpus. All the original instances are sentence-tokenized and the punctuation was removed. The details of creating the datasets are as follows:

- FrameNet: Non per-target type. We get FrameNet annotated instances through Semlink. If one FrameNet frame label contains more than 300 instances, we divide it proportionately: 70%, 20% and 10%. Then we respectively accumulate the three parts by each frame label to create the training, test and validation set.

- PropBank: Per-target type. The creation process is same as FrameNet except that we finally get training, test and validation set for each target and the cutoff is set to 70 instead of 300.

- PDEV: Same as PropBank but with the cutoff set to 100 instead of 70.

Since the performance of our model is almost decided by the training data we empirically choose the cutoff above to keep the instances of each label enough. Summary statistics of the above datasets are in Table 2.

3.2 Models and Training

We compare our model with the following baselines.:

[1] The current version of the Semlink project has some problems to get the right position of targets in WSJ section of Penn Treebank. Instead, we use annotations of PropBank corpus, also annotated in WSJ section of Penn Treebank, to index targets.

[2] http://pdev.org.uk/

Sentences	Frame Names
In Moscow they **kept** asking us things like why do you make 15 different corkscrews	Activityongoing
It said it has taken measures to **continue** shipments during the work stoppage.	Activityongoing
But the Army Corps of Engineers expects the river level to **continue** falling this month.	Processcontinue
The oil industry's middling profits could **persist** through the rest of the year.	Processcontinue

Table 1: Non per-target examples. Frames are from FrameNet and the target words are in bold.

	FrameNet	PropBank	PDEV
Per-target	No	153 targets	407 targets
Train	41206	31212 (204)	152218 (374)
Test	11762	8568 (56)	42328 (104)
Valid.	5871	4131 (27)	20350 (50)
Frame	33	443 (2.89)	2197 (5.39)
Words/sent.	23	23	12

Table 2: Summary statistics for the datasets. The average numbers per target are shown in the parentheses for per-target resources.

- MF: The most frequent (MF) method selects the most frequent semantic frame label seen in training instances for each instance in the test dataset. MF is actually a strong baseline for per-target dataset because we observed that most targets have one main frame label.

- Target-Only: For FrameNet dataset, we use Target-Only method: if the target in the test instance has a unique frame label in the training data we give this frame label to current test instance; if the target has multiple frame labels in the training data we select the most frequent one in these labels; if the target is not seen in the training data, we select the most frequent label from the whole training data. This baseline is especially for FrameNet because we observed that each frame label has a set of targets but only a few targets have multiple frame labels. It may be easy to predict the frame label for test instances only according to the target.

- LSTM: The standard LSTM model.

- MaxEnt: The Maximum Entropy model. We use the Stanford CoreNLP module [3] to ex-

[3] http://stanfordnlp.github.io/CoreNLP/

tract features for MaxEnt toolkit [4]. All dependents related to the target, their POS tags, dependency relations, lemmas, NE tags and the target itself will be extracted as features.

The number of the iterations for MaxEnt is decided by the validation set. For simplicity, we set the learning rate to 1.0 for TRNN and LSTM. The number of hidden units is tested on validation data with the values {35, 45, 55} for per-target resource and {80, 100, 120} for non per-target resource. We use the publicly available word2vec vectors, a dimensionality of 300, that were trained through the GloVe model (Pennington et al., 2014) on Wikipedia and Gigaword. For words not appeared in the vector model, their word vectors are all set to zero vectors. We train these models by stochastic gradient descent with minibatches. The minibatch is set to 10 for per-target resource and 50 for non per-target resource. We keep the word vectors static since no obvious improvement has been observed. Training will stop when the zero-one loss is zero over training data.

3.3 Results

The results of the above datasets are in Table 3. Target-Only gets very high scores on FrameNet dataset. FrameNet dataset has 55 targets which has multiple frame labels in the training data and these targets have 1981 instances in the test data. We get 0.769 F-score on these instances and 0.393 F-score on 64 unseen targets with 77 test instances. This can be the extreme case that the main feature for the correct frame is the target itself. Despite this simple fact, standard LSTM performs very badly on FrameNet. The main reason is that sentences in FrameNet dataset are too long and standard LSTM can not learn well due to the large number of irrelevant words that appear in long sentences. To show this, we select the size of truncation window for original FrameNet sentences and we get the best size of 5 on validation data with each 2 words surrounding the target. Finally, we get 0.958 F-score on FrameNet test data which is still lower than TRNN on full sentences. As for PropBank and PDEV dataset, we train one model for each target so the final F-score is the average of all targets. However, the number of training instances per target is limited. TRNN will usually not perform well when it tries to learn some frames which consist of many different concepts and especially when the frame has a few training instances. Considering the sentence 4 of Table 4 as an example, it is difficult to TRNN to learn what is 'Activity' in the correct frame because this concept is huge. TRNN may need lots of data to learn something related to this concept. However, this correct frame only has 6 instances in our training data. The second reason of TRNN's failure is lack of knowledge due to unseen words in test data. The sentence 1 of Table 4 shows TRNN will make the right decision since we observe that it has seen the word 'cow' in the training data and knows this word belongs to the concept 'Animate or Plant' in the correct frame. But TRNN does not know the word 'Elegans' in sentence 3 so it usually selects the most frequent frame seen in the training data. However, in many cases, the unseen words can be captured by well trained word embeddings as the sentence 2 shows where 'ducks', 'chickens' and 'geese' are all unseen words.

Models	FrameNet	PropBank	PDEV
MF	0.38	0.78	0.61
Target-Only	0.911	-	-
MaxEnt	0.829/125	0.874/30	0.704/10
LSTM	0.55/80	0.78/35	0.72/55
TRNN	**0.962/100**	**0.887/35**	**0.794/55**

Table 3: Results on several semantic frame resources. The format of cell value is "F-score/hidden unit" for TRNN and LSTM and "F-score/iteration" for MaxEnt toolkit.

3.4 CPA Experiment

Corpus Pattern Analysis (CPA) is a new technique for identifying the main patterns in which a word is used in text and is currently being used to build the PDEV resource as we mentioned above. It is also a shared task in SemEval-2015 task 15 (Baisa et al., 2015). The task is divided into three subtasks: CPA parsing, CPA clustering and CPA lexicography. We only introduce the first two related subtasks. CPA parsing aims at identifying the arguments of the target and tagging predefined semantic meaning on them; CPA clustering clusters the instances to obtain CPA frames based on the result of CPA parsing. However, the first step results seem unpromising (Feng et al., 2015; Mills and Levow, 2015; Elia, 2016) which will influence the process of obtaining CPA frames. Since our model can be applied on sentence-level input without feature extraction we can directly evaluate

[4] https://github.com/lzhang10/maxent

ID	Sentences	Frame Prediction	True Frame
1	One of the farmer's cows had **died** of BSE raising fears of cross-infection...	Same with true frame	Animate or Plant dies
2	One of the farmer's ducks|chickens|geese had **died** of BSE raising fears of cross-infection...	Same with true frame	Animate or Plant dies
3	Elegans also in central America **die** of damping off as a function of distance	Human dies ((Time Point)(Location)(Causation) (at Number or at the age of or at birth or earlage))	Animate or Plant dies
4	Indeed, the MEC does not **advise** the use of any insecticidal shampoo for...	Human 1 or Institution 1 advises Human 2 or Institution 2 to-infinitive	Human or Institution advises Activity

Table 4: Case study for CPA frames. The target words are in bold.

our model on CPA clustering. Unfortunately, the datasets provided by CPA clustering is a per-target resource for our model and the targets in training and test set are not the same. Since this task is not limited to use extra resources, we use the training set of FrameNet, a type of non per-target, mentioned in section 3.1 to solve this problem. The hyper parameters are the same as before. CPA clustering is evaluated by B-cubed F-score, a metric for clustering problem, so we do not need to convert the FrameNet frame label to CPA frame label. The result is in Table 5. All the models are supervised except for baseline and DULUTH. Feng et al. (2016) used the MLP to classify fixed-length local text of the target based on distributed word embeddings. But the representation of the target's context is simply constructed with concatenated word embeddings and the length of local context has to be chosen manually. Besides, MLP may fail to train or predict well when some key words are out of its input window.

System	B-cubed F-score
BOB90(Best in SemEval 2015)	0.741
SemEval 2015 baseline	0.588
DULUTH	0.525
Feng et al. (2016)	0.70
This paper	**0.763**

Table 5: Results on Microcheck dataset of CPA clustering.

3.5 Word Sense Disambiguation Experiment

Finally, we choose Word Sense Disambiguation (WSD) task to extend our experiment. As our benchmark for WSD task, we choose English Lexical Sample WSD tasks of SemEval-2007 task 17 (Pradhan et al., 2007). We use cross-validation on the training set and we observe the model performs better when we update the word vectors which is different from the preceding experimental setup. The number of hidden units is set to 55. The result is in Table 6. The rows from 4 to 6 come from Iacobacci et al. (2016). They integrate word embeddings into IMS (It Makes Sense) system (Zhong and Ng, 2010) which uses support vector machine as its classifier based on some standard WSD features and they get the best result; they use an exponential decay function, also designed to give more importance to close context, to compute the word representation, but their method need manually choose the window size of the target word and one parameter of their exponential decay function. Both with word vectors only, our model is comparable with the sixth row.

System	F-score
Rank 1 system in SemEval 2007	0.887
Rank 2 system in SemEval 2007	0.869
IMS (2010)	0.879
IMS + word vectors (2016)	**0.894**
IMS + word vectors only (2016)	0.880
This paper	0.886

Table 6: Result on Lexical Sample task of SemEval-2007 task 17

4 Conclusion

In this paper, we describe an end-to-end neural model to target-specific semantic frame labeling. Without explicit rule construction to fit for some specific resources, our model can be easily applied to a range of semantic frame resources and similar tasks. In the future, non-English semantic frame resources can be considered to extend the coverage of our model and our model can integrate the best features explored in the state-of-the-art work to see how many improvements our model can make.

Acknowledgments

We would like to thank the anonymous reviewers and Li Zhao for their helpful suggestions and comments. The work was supported by the National High Technology Development 863 Program of China (No.2015AA015409).

References

Omri Abend, Roi Reichart, and Ari Rappoport. 2008. A supervised algorithm for verb disambiguation into verbnet classes. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, pages 9–16.

Vít Baisa, Jane Bradbury, Silvie Cinkova, Ismail El Maarouf, Adam Kilgarriff, and Octavian Popescu. 2015. Semeval-2015 task 15: A cpa dictionary-entry-building task. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Association for Computational Linguistics, Denver, Colorado, pages 315–324. http://www.aclweb.org/anthology/S15-2053.

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, pages 86–90.

Ismaıl El Maarouf and Vıt Baisa. 2013. Automatic classification of patterns from the pattern dictionary of english verbs. In *Joint Symposium on Semantic Processing.*.

Ismail El Maarouf, Jane Bradbury, Vít Baisa, and Patrick Hanks. 2014. Disambiguating verbs by collocation: Corpus lexicography meets natural language processing. In *LREC*. pages 1001–1006.

Francesco Elia. 2016. Syntactic and semantic classification of verb arguments using dependency-based and rich semantic features. *arXiv preprint arXiv:1604.05747* .

Yukun Feng, Qiao Deng, and Dong Yu. 2015. Blcunlp: Corpus pattern analysis for verbs based on dependency chain. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Association for Computational Linguistics, Denver, Colorado, pages 325–328. http://www.aclweb.org/anthology/S15-2054.

Yukun Feng, Yipei Xu, and Dong Yu. 2016. An end-to-end approach to learning semantic frames with feedforward neural network. In *Proceedings of the NAACL Student Research Workshop*. Association for Computational Linguistics, San Diego, California, pages 1–7. http://www.aclweb.org/anthology/N16-2001.

Patrick Hanks. 2012. How people use words to make meanings: Semantic types meet valencies. *Input, Process and Product: Developments in Teaching and Language Corpora* pages 54–69.

Karl Moritz Hermann, Dipanjan Das, Jason Weston, and Kuzman Ganchev. 2014. Semantic frame identification with distributed word representations. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Baltimore, Maryland, pages 1448–1458. http://www.aclweb.org/anthology/P/P14/P14-1136.

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

Matthew Honnibal and Tobias Hawker. 2005. Identifying framenet frames for verbs from a real-text corpus. In *Proceedings of Australasian Language Technology Workshop*.

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2016. Embeddings for word sense disambiguation: An evaluation study. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. pages 897–907.

Edward Loper, Szu-Ting Yi, and Martha Palmer. 2007. Combining lexical resources: mapping between propbank and verbnet. In *Proceedings of the 7th International Workshop on Computational Linguistics, Tilburg, the Netherlands*.

Chad Mills and Gina-Anne Levow. 2015. Cmills: Adapting semantic role labeling features to dependency parsing. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Association for Computational Linguistics, Denver, Colorado, pages 433–437. http://www.aclweb.org/anthology/S15-2075.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics* 31(1):71–106.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*. volume 14, pages 1532–1543.

Sameer S Pradhan, Edward Loper, Dmitriy Dligach, and Martha Palmer. 2007. Semeval-2007 task 17: English lexical sample, srl and all words. In *Proceedings of the 4th International Workshop on Semantic Evaluations*. Association for Computational Linguistics, pages 87–92.

Karin Kipper Schuler. 2006. *VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon*. Ph.D. thesis, University of Pennsylvania. http://verbs.colorado.edu/ kipper/Papers/dissertation.pdf.

Hongyan Zhao, Ru Li, Sheng Zhang, and Liwen Zhang. 2016. Chinese frame identification with deep neural network 30(6):75.

Zhi Zhong and Hwee Tou Ng. 2010. It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the ACL 2010 System Demonstrations*. Association for Computational Linguistics, pages 78–83.