UNIST SAIL System for TAC 2017 Cold Start Slot Filling

Seongwoo Lim[*], **Sunjae Kwon**[*], **Sehyun Lee**, and **Jaesik Choi**

School of Electrical and Computer Engineering
Ulsan National Institute of Science and Technology
{seongwoo, soon91jae, tkdtkdghkt, jaesik}@unist.ac.kr

Abstract

We describes the UNIST SAIL Slot Filling System for TAC 2017 Cold Start Slot Filling (CSSF) task. Our system uses Pattern-based relation extraction (Distant supervision based model) and Convolutional Neural Network (Deep Learning based End-to-End model). The first model achieved 0.1474 on Average Precision (AP), the second model achieved 0.1352 on AP for TAC 2017 CSSF task.

1 Introduction

Knowledge Base is one way to represent and store knowledge. The knowledge is defined as the relation between entities with triplet form (e.g. ("Barack Obama", per:spouse, "Michelle Obama")).

Because it is practically hard and expensive to populate the Knowledge base, automatic knowledge base population from unstructured text is important challenge in natural language process.

From 2009, National Institute of Standards and Technology has been annually opening tasks for knowledge base population. In this paper, we describe the UNIST SAIL System for TAC 2017 Cold Start Slot Filling (CSSF) task.

The Slot Filling task is a kinds of relation extraction problem. The query consists of entity, e_{query} and pre-defined relation, r. The system should response 1) the entity, e_{answer}, which has relation r with e_{query} 2) the sentence representing the relation r between e_{query} and e_{answer} from unstructured text. Cold Start means that the system should populate knowledge base from empty knowledge base, in other words, it is not allowed to response the query by searching knowledge base.

Our system uses Pattern-based relation extraction (Distant supervision based model) and Convolutional Neural Network (Deep Learning based End-to-End model). Distant supervision based model extracts feature from unstructured text by using the position of entities. The model maps sentence to feature space by Bag-Of-Feature. Next, the model predicts the relation between entities by using Multi-Layer Perceptron to sentence embedding. Deep Learning based End-to-End model consist of two sub models 1) Bi-directional Long Short Term Memory-Conditional Random Fields (BLSTM-CRF) based candidate extractor 2) Piecewise Convolutional Neural Network (PCNN) based re-ranker. Both of sub-models jointly optimized combining loss function of each model. Finally, We made rule based for the 'org:website' class.

2 Model

2.1 Distant supervision based model

We developed distant supervision (Mintz et al., 2009) based feature extraction model. The model consists of distant supervision based sentence feature extractor and Multi-Layer Perceptron (MLP) based sentence classifier. We used Stanford CoreNLP (Manning et al., 2014) for Part-Of-Speech (POS) tagging, Lemmatization, Named Entity Recognition (NER) and Dependency Parsing. To extract candidate from sentence, the NER re-

[*] Both authors contributed equally to this work.

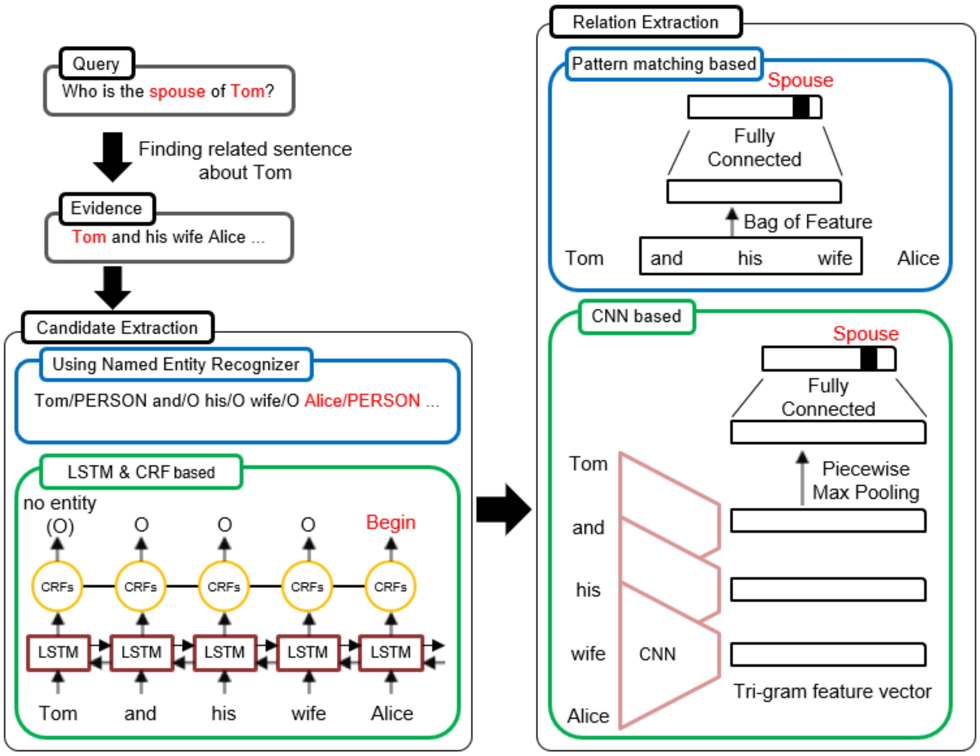

Figure 1: Our system for TAC 2017 CSSF task. Blue is Distant supervision based model, Green is Deep Learning based End-to-End Model.

sult and our rule (section 2.3) were used. We split the sentence as 3 part. First, the front phrase of given entity and candidate (entity pair). Second, the phrase between entity pair. Finally, the back phrase of entity pair. The extracted features consist of each part of sentence with the POS, Lemma, NER, Dependency path from the entity pair. We clustered some related relation (e.g. org:founded_by, org:top_memebers_employees). The feature sets for each relation cluster was collected from our training data (KBP 2012-2015 dataset, Angelis dataset and manually labeled dataset). We dropped the features which satisfy both of two conditions. First, the feature did not appear in positive data. Next, the feature appeared less than 3 times. The bag of feature vector that consists of the extracted features is used as input of MLP. The output dimension of MLP is the number of relations in relation cluster. The output of MLP represents the probability whether the sentence represents the relation for entity pair or not.

2.2 Deep Learning based End-to-End model

Apart from 1.1, we developed a deep learning end-to-end model based on the supervised learning method. The proposed model consists of a candidate answer extractor and a re-ranker.

2.2.1 Candidate Answer Extractor

The candidate answer extractor recognizes the pattern of the entity that related to the given entity. We formulate this as a sequence labeler that labeling all the token X as a corresponding tag sequence Y. For example, in the Figure 1 input sentence “Tom and his wife Alice” can be tagged with [‘O’, ‘O’, ‘O’, ‘O’, ‘Begin’]. At that time, ‘O’ indicates it is not the candidate answer and ‘Begin’ means this token is a beginning of the candidate answer.

$$X = (x_1, x_2, ..., x_k) \quad (1)$$

$$\mathbf{y} = (y_1, y_2, ..., y_k) \quad (2)$$

We used Bi-directional Long Short Term Memory Conditional Random Fields(Bi-LSTM-CRFs) for our candidate answer extractor (Lample et al., 2016).

2.2.2 Re-ranker

The re-ranker assign the relation scores between the candidate answer and the given entity. Re-ranker is developed by Piecewise Convolutional Neural Network (PCNN) (Zeng et al., 2015).

There can be several relations between the given entity and the candidate answer at the same time. For example, 'org:employee' and 'org:top_employee' relation are found simultaneously between given entity 'IBM' and candidate answer 'Ginni Rometty' in the sentence "IBM chief Ginni Rometty puts emphasis on responsible use of data". Therefore, categorizing the relation with the highest performance as the correct answer, such as simple multi-class classification case, causes performance degradation. In order to ameliorate this problem, the final answer is determined by thresholding the re-ranker score. In this case, if the confidence scores of several classes exceed thereshold together, then there is more than one relationship between the given entity and candidate answer entity. The threshold is learned by the ridge regression of the validation set 2015 KBP corpus following the method of (Xu et al., 2017).

2.2.3 Multi-task Optimization of Candidate Answer Extractor and Re-ranker

Recently, some studies have constructed a deep learning model using multi-task learning method (Ruder, 2017). Multi-task learning is a method of simultaneously learning different models that perform different tasks. It is known to be advantageous from the viewpoint of generalization of models. The candidate answer extractor and re-ranker are optimized simultaneously with a eq.5.

$$loss_{cand} = -log(P(\hat{Y}|X, e1)) \quad (3)$$

$$loss_{rerank} = \sum_{i=0}^{n} CE(\hat{y}, re - ranker(X, e1, e2_i)) \quad (4)$$

$$loss_{multi} = \alpha \times loss_{cand} + (1-\alpha) \times loss_{rerank} \quad (5)$$

Eq.3 is the loss function of the candidate answer extractor. Where X is the input sequence and $\hat{Y}$ is the answer tag sequence for the input sentence X and given entity $e1$. We maximize the conditional probability of $\hat{Y}$ to given X with negative log likelihood loss function. Eq.4 is the loss function for optimizing the re-ranker. We jointly optimize all n entities $e2$ in the sentence related to the given entity with categorical cross entropy (CE) loss. Note that, if there is no entity related to the given entity in the sentence then we set loss as a zero. Finally, linear combination of Eq.3 and Eq.4 yields the final loss function eq.5.

2.2.4 Model specification

There are 4 features composed input vectors. First of all, we used 100 dimension word embedding learned by wang2vec algorithm (Ling et al., 2015). Secondly, we used 25dimension character based word embedding which concatenated vector of character-level BLSTM of each word (Lample et al., 2016). Third one is 10 dimension part-of-speech embedding. Final one is 10 dimension NER embedding. The input layer is shared both candidate answer extractor and re-ranker.

For the candidate answer extractor, we set the hidden dimension as 100 dimension. In addition, hidden dimension of re-ranker is set to 100 dimension and window size of re-ranker is set to 3. Dropout regularization is used to prevent over-fitting problem (Srivastava et al., 2014). We apply dropout layer on the input layer.

2.3 Rule based model for website class

Root domain has an information of what website owner operates. To make it memorable, it usually consists of keywords which are related with their business. In addition, to make it simple and short, it often uses an abbreviation of business's name. In this regard, we gave an score for similarity between root domain and given entity. We use regular expression to extract root domain from website address. After then, we calculate similarity score of root domain and given entity utilizing 1) Exact string matching, 2) Substring matching, 3) Abbreviation string matching and 4) Ratcliff/Obershelp(Black, 2004) algorithm. Finally, We set the threshold and submit the answer website re-

	Hop0			Hop1			All		
	$Prec.$	$Rec.$	F_1	$Prec.$	$Rec.$	F_1	$Prec.$	$Rec.$	F_1
UNIST_SAIL_ENG_1	0.125	0.381	0.159	0.031	0.082	0.043	0.088	0.265	0.114
UNIST_SAIL_ENG_2	0.205	0.199	0.187	0.000	0.000	0.000	0.126	0.122	0.114

Table 1: The KBP 2017 CSSF result of UNIST SAIL team on the macro F_1-score

late to given entity. This rule based web class model is used as a part of both 1.1 and 1.2 model.

	Hop0	Hop1	All
UNIST_SAIL_ENG_1	0.212	0.010	0.147
UNIST_SAIL_ENG_2	0.182	0.000	0.135

Table 2: The KBP 2017 CSSF result of UNIST SAIL team on the mean average precision criterion

3 Result

For TAC 2017 CSSF track, we submitted 2 kinds of submission.

UNIST_SAIL_ENG_1 Distant supervision based model + Rule based model for website class

UNIST_SAIL_ENG_2 Deep Learning based End-to-End model + Rule based model for website class

There are two criteria for the evaluation of model performance; F_1 score and Mean average precision (MAP). Table 1 demonstrates the performance on the F_1 Score and Table 2 shows the performance on the MAP. Experimental results show that our deep learning based model had higher performance than distant supervision based model in F_1 score criterion. Meanwhile, distant supervision based model achieved higher performance than deep learning based model in the MAP criterion.

Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) grant (NRF2014R1A1A10026) and Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (No.2017-0-01779, A machine learning and statistical inference framework for explainable artificial intelligence).

References

Paul E Black. 2004. Ratcliff/obershelp pattern recognition. *Dictionary of Algorithms and Data Structures*, 17.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *CoRR*, abs/1603.01360.

Wang Ling, Chris Dyer, Alan Black, and Isabel Trancoso. 2015. Two/too simple adaptations of word2vec for syntax problems. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

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.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Guanghao Xu, Hyunjung Lee, Myoung-Wan Koo, and Jungyun Seo. 2017. Convolutional neural network using a threshold predictor for multi-label speech act classification. In *Big Data and Smart Computing (BigComp), 2017 IEEE International Conference on*, pages 126–130. IEEE.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Emnlp*, pages 1753–1762.