SAIL (UNIST) team Participation in KBP 2016 Cold Start Slot Filling

Seongwoo Lim and **Jiyeon Han** and **Kyowoon Lee** and **Jaesik Choi**
Ulsan National Institute of Science and Technology
Ulsan, Republic of Korea
{seongwoo,jiyeon,leekwoon,jaesik}@unist.ac.kr

Abstract

Building a Knowledge Base (KB) from unstructured text is one of challenges in natural language processing. From 2009, National Institute of Standards and Technology (NIST) has been annually opening challenge, Knowledge Base Population (KBP)[1], for this work.

In this paper, we focus on automatic knowledge base population by Cold Start Slot Filling (Cold Start SF)[2] which is a kind of NIST KBP. We will use Distant Supervision (DS), widely used to this problem, to extract feature from unstructured text and map sentence to feature space. We also use neural network to extend representation power of feature. Some cases, we can not find the answer using DS. In this case, we use bidirectional Long Short-Term Memory (LSTM) to find the answer.

1 Introduction

Knowledge Base (KB) such as YAGO(Mahdisoltani et al., 2014), Wikidata[3] is a kind of database which represent relation between data. Generally KB is represented as triplet like *Spouse*("Barack Obama", "Michelle Obama") which means that "Barack Obama" has *Spouse* relation to "Michelle Obama". KB is used to various artificial intelligence system (e.g. IBM Watson) as important part. Building and maintaining KB by hand is practically hard. For this reason, building a KB from unstructured text such as news article is important technology.

NIST TAC KBP, the challenge for this work, is consist of 5 tracks (Cold Start SF/KB, Entity Discovery and Linking, Validation/Ensembling, Event, Source-and-Target Belief and Sentiment Evaluation). In this paper, we focus on Cold Start SF track. SF is a kind of relation extraction. Given text data (e.g. New York Times 2013) and query (e.g. *Spouse*("Barack Obama", ?)), we should find answer (e.g. "Michelle Obama") and justification (e.g. "President Barack Obama and first lady Michell Obama."). Cold Start means that we should start from empty KB, in other words, we can not search relation from our KB. If we build the system for Cold Start SF, we can automatically build and extend a KB by using only text data. It can be used to reduce time of doctor and lawyer(Gordon, 2003). Many research institute such as Stanford University(Angeli et al., 2015), New York University(He and Grishman,), University of Massachusetts Amherst(Roth et al., 2015), Carnegie Mellon University(Kisiel et al.,) have been participating in NIST TAC KBP Cold Start SF.

From this year, NIST KBP starts cross-lingual (English, Chinese, Spanish) Cold Start SF. We only participate in English Cold Start SF. NIST KBP 2016 English Cold Start SF is consist of around 30000 document (around 922,663 sentence), 1350 query (for hop 0) and 65 relation. The criteria for NIST KBP Cold Start SF are precision[5], recall[6] and

[1]http://tac.nist.gov/2016/KBP/index.html

[2]http://tac.nist.gov/2016/KBP/ColdStart/index.html

[3]https://www.wikidata.org/

[4]https://tac.nist.gov/2015/KBP/ColdStart/guidelines/TAC_KBP_2015_Slot_Descriptions_V1.0.pdf

[5]Precision = (# of correct output) / (# of output)

[6]Recall = (# of correct output) / (# of answer in answer

Table 1: Example of relations in NIST KBP Cold Start SF[4]

Relation	Description
per:title	Official or unofficial name(s) of the employment or membership positions of the assigned person.
per:spouse	The spouse(s) of the assigned person
org:top_members_employees	The persons in high-level, leading positions of the assigned organization.
org:city_of_headquaters	Location of the headquarters of the assigned organization at the city, town, or village level.

F1[7] score. F1 score is main criteria for NIST KBP Cold Start SF.

In this paper, we will use Distant Supervision (DS) based method to extract feature from unstructured text and map sentence to feature space. We also use neural network to extend representation power of feature. Some cases, we can not find the answer using DS based method (e.g. When we can not find candidate from sentence). In this case, we use bidirectional Long Short-Term Memory (LSTM) to find the answer.

2 Relate Works

In this section, we will explain approach to solve SF. First, relation extraction using Distant Supervision, second, Multi-Instance Multi-Label Learning, third, Recurrent Neural Network.

2.1 Relation extraction using Distant Supervision (DS)

Relation extraction using DS(Mintz et al., 2009) is general and instinctive way to solve SF(Fan et al., 2012; Mintz et al., 2009; Zelenko et al., 2003). DS assumption is that if two entity have a relation and there are sentences which include two entity, then sentences have high probability to represent relation. For example, if we know relation *Capital*("South Korea","Seoul"), then the sentence, "Seoul is the capital and largest metropolis of the South Korea.", have high probability to represent relation, *Capital*("South Korea","Seoul").

Relation extraction using DS use the pattern (feature) of sentence which represent relation (Figure 1-(a)). In previous *Capital*("South Korea","Seoul") example, we can extract feature, "B is the capital and largest metropolis of the A.". If we find the sentence replaced A, B to other entity pair C,D, "D is the capital and largest metropolis of the C.", then we can extract relation *Capital*("C","D") from the sentence. We will define the type of feature, for example, 1)words between entity pair 2)words on path between entity pair in dependency tree of sentence are widely used as feature.

Extracted feature by using DS is not always represent relation (Figure 1-(b)). Entity pair can have more than one relation (e.g. spouse and family). Extremely, entity pair can appear in one sentence by chance. For example, there are not only *Capital*("South Korea","Seoul") relation, but also *Cities*("South Korea","Seoul"). Although the sentence, "This meeting is open in the Seoul, South Korea." include both "South Korea" and "Seoul", this sentence doesn't represent *Capital*("South Korea","Seoul") relation. To solve this problem, 1)use active learning to decrease noise(Angeli et al., 2014b; Sterckx et al., 2014) 2)use Statistical Relational Model to add inference rules(Niu et al., 2012; Angeli et al., 2014a) were suggested. Another way to use DS is making negative example using DS(Zhang, 2015). For example, if we know that *Capital*("South Korea","Ulsan") is wrong, then every sentence which include both "South Korea" and "Ulsan" will not represent *Capital* relation.

2.2 Multi-Instance Multi-Label Learning (MIML Learning)

As we mentioned in 2.1, the entity pair can have more than one relation. MIML Learning(Surdeanu et al., 2012) is method to learn all relation between entities. To extract relation using MIML, first, the system finds all sentence which include

sheet)

[7] F1 = 2 * Precision * Recall / (Precision + Recall)

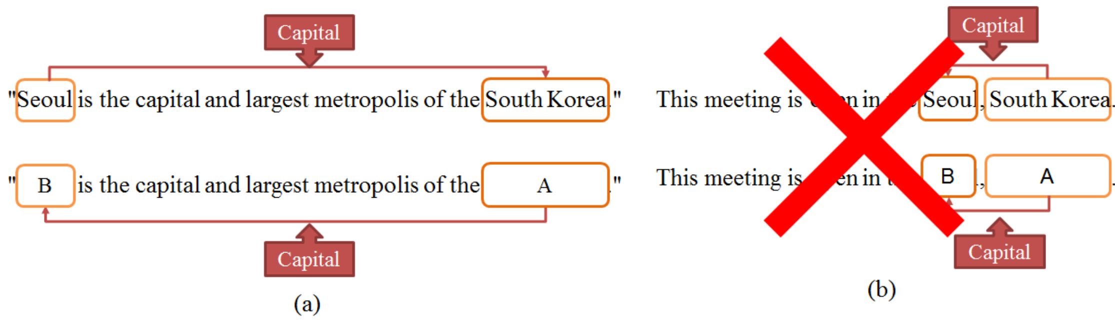

Figure 1: Example of feature extraction by DS

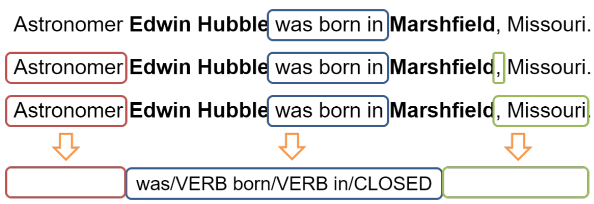

Figure 2: Example of features used for relation extraction using DS.

entity pair. Second, the system use every sentence as input to $z\_classifier$. After second process, the system make $relation\_vector$ by using all output of $z\_classifier$. Each element of $relation\_vector$ represent $relation_i$. Next, each element of $relation\_vector$ was used as input to $y\_classifier_i$. The $y\_classifier_i$ makes output that the entity pair has $relation_i$ relation or not.

2.3 Recurrent Neural Network

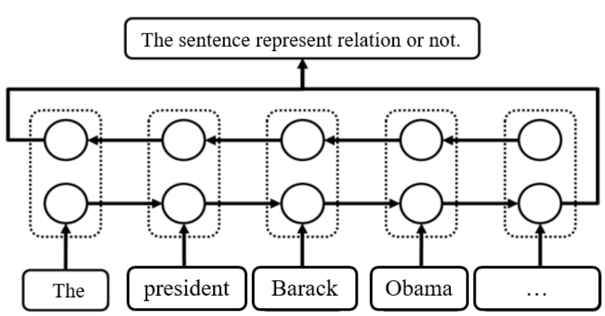

Figure 3: Example of classify sentence using BRNN

Both relation extraction using DS and MIML

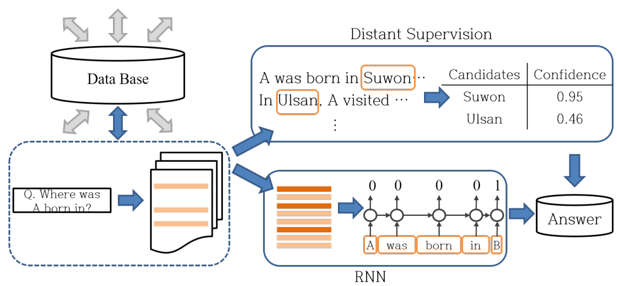

Figure 4: This figure describes the whole process of our system.

Learning extract the relation between entity pair, in other words, to solve SF by using these ways, we find candidate of answer first. When the type of answer is person or organization, we can find candidate by using Named Entity Recognizer. When the type of answer is job or a kind of crime, finding candidate of answer is hard. In these case, Using the Bi-directional Recurrent Neural Network (BRNN) was suggested(Vu et al., 2016). They use BRNN to classify whether the sentence represent the relation or not. For example (Figure 3), if the query is *per:title*(”Barack Obama”, ?), then find all sentence which include ”Barack Obama” and then use every sentence as input to BRNN. BRNN makes binary output which means that the sentence represent *per:title* relation of ”Barack Obama”. And then find job candidate using other algorithm.

3 Model

In this part, we will introduce our system for KBP 2016 Cold Start SF. The procedure is as following (Figure 4). First, find all sentences which contain the entities of the given query. Second, if we can find candidates from the sentence by using Named Entity Recognizer, then we find candidates and use DS based approach. If we can not find candidates from the sentence, use Neural Networks approach. After finding the answer, we filter the answer using properties of the query (e.g. single-value slot, the type of answer is city).

3.1 Distant supervision based approach

One of necessary conditions for relation extraction using DS is enough data to extract feature (pattern) of relation. However, the more data is used, the more features (with noise) are extracted. When we use too many feature, it require too much computation during training and test. Finally it can cause slowing down and performance degradation. To avoid this problem, we deleted features which the number of appearances is less than *k*.

One of characteristics of natural language is that same meaning can be expressed in several ways. Therefore the recall of relation extraction using DS is low.

To solve above problems, we used separated feature rather than combined feature. In figure 2, relation extraction using DS used word sequence before/between/after entity pair and part of speech of the word sequence (e.g. "Astronomer (entity1) was/VERB born/VERB in/CLOSED (entity2) Marshfield") as one feature. We used each part of pattern (e.g. "was born in") as one feature (figure 5)

We used DS based approach to classify whether the candidate is the answer or not when we can find candidates from the sentence. We extract features (Table 1) from the sentence by using Natural Language ToolKit[8] , Stanford NLP tools (POS tagger[9], dependency parser[10] and Named Entity Recognizer[11]) and WordNet[12]. We used 2-layer perceptron to classify whether the candidate is the answer or not when we can find candidates from the sentence.

Type of feature	Feature
Based on location of feature	Words between entity pair
	Words after entity pair
	Words before entity pair
	Words on dependency path between entity pair
	Words on dependency path after entity pair
	Words on dependency path before entity pair
Based on property of feature	Lemma of words
	POS of words
	Named entity tag of words
Others	The number of words between entity pair
	Order of given entity and the candidate

Table 2: Table of feature

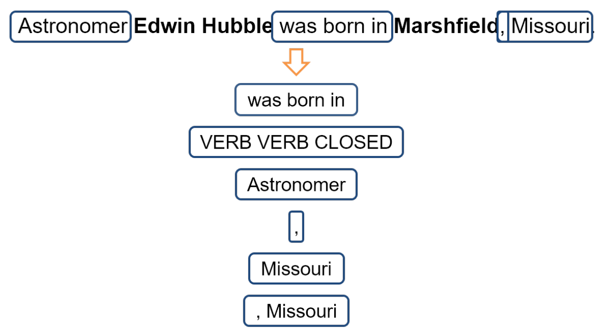

Figure 5: Example features we used for DS based approach.

3.2 Neural network

As some relations are hard to find answer candidates using Named Entity Recognition, it is hard to apply Distant Supervision to those relations. To solve this problem we used two Neural Networks. The first Neural Network classify weather the input sentence represents given relation or not. The second one finds the answer words in the sentence that we found in the first Neural Network.

[8]http://www.nltk.org/

[9]http://nlp.stanford.edu/software/tagger.shtml

[10]http://nlp.stanford.edu/software/stanford-dependencies.shtml

[11]http://nlp.stanford.edu/software/CRF-NER.shtml

[12]http://wordnet.princeton.edu

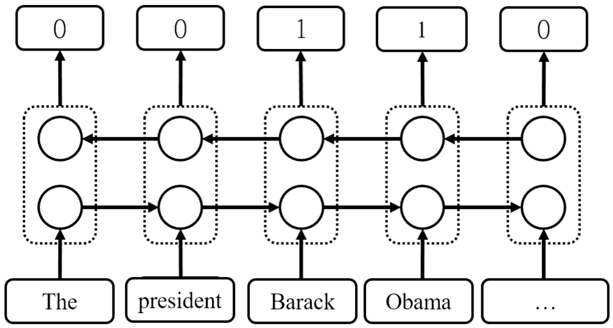

Figure 6: Find the answer words using BRNN

As the input to the Neural Networks, we used the following vectors appended to the Word2Vec vector of each word in a sentence.

one-hot encoding of POS/one-hot encoding of NER/whether the input is in Word2Vec vocabulary (1) or not (0)/whether the input is a 'null' (1) or a word (0).

For the first Neural Network, we used the Bidirectional LSTM (Long-Short Term Networks) with 10 words forward the keyword and 10 words backward the keyword as the input. We added 1 Fully-Connected layer on top of the Bidirectional LSTM with two nodes, each represents weather the input sentence is representing the relation or not.

For the second Neural Network (figure 6), we used the same input as in the first Neural Network to the Bidirectional Neural Network. We also added 1 Fully-Connected layer with 20 nodes, each represents the probability of the corresponding input word to be the answer.

4 Experiments

Distant Supervision and Neural Networks are in common in that they both use sentences after preprocessing such as POS or NER tagging. As it takes about 10 seconds to process one sentence in our experimental environment, the total expected time for preprocessing all TAC KBP 2016 data is 10 (sec) $\times$ 922,633 (the number of sentences in TAC KBP 2016 data set) $\times$ 3,600 (sec) = 2,562 (hour). To reduce preprocessing time we built a distributed system with 8 servers.

We used MongoDB so that servers can communicate with each other. Client servers could access to MongoDB of the host server and compute in parallel. The distributed procedure is as following (figure 4). First, a client server read a query and check from the database if the query is being processed or already processed. If the query is being processed or already processed, read the next query. Otherwise, find answers for the query. To find answers for the query, a client server need to find sentences that contain the keyword. A client server check from the server if the keyword is processed before or not. If the keyword is not processed before, preprocess the sentences that contain the keyword and upload the preprocessed sentences to the database. If the keyword is processed before, get preprocessed sentences from the database.

We could process 6,679 sentences per hour making use of this distributed system. This could be possible as we reduced unnecessary repetition of processing same keywords and processing queries in parallel.

5 Results

For KBP 2016 Cold Start SF, we submitted 4 kind of submission.

UNIST_SAIL_ENG_1 Only use DS based approach
UNIST_SAIL_ENG_2 Use DS based approach + Bidirectional LSTM
UNIST_SAIL_ENG_3 Only use DS based approach + Set the rule by hand
UNIST_SAIL_ENG_4 Use DS based approach + Set the rule by hand + Bidirectional LSTM

In first and third submission, we ignore query when we can not find candidate. The meaning of "Set the rule by hand" is that we set some training data as negative. For example, "UNIST in Ulsan" could be justification of "org:city_of_headquarters(UNIST,?)". It is hard to decide with only "UNIST in Ulsan", because headquarters of UNIST could be located in other place. **UNIST_SAIL_ENG_4** perform best out of our submissions (figure 7).

References

Gabor Angeli, Sonal Gupta, Melvin Jose, Christopher D Manning, Christopher Ré, Julie Tibshirani, Jean Y Wu,

	Hop0			Hop1			All		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
UNIST_SAIL_ENG_1	0.0870	0.0275	0.0417	0.0000	0.0000	0.0000	0.0087	0.0182	0.0118
UNIST_SAIL_ENG_2	0.0751	0.0350	0.0477	0.0000	0.0000	0.0000	0.0751	0.0232	0.0354
UNIST_SAIL_ENG_3	0.0958	0.0375	0.0539	0.0000	0.0000	0.0000	0.0958	0.0248	0.0394
UNIST_SAIL_ENG_4	0.0792	0.0624	0.0698	0.0000	0.0000	0.0000	0.0792	0.0414	0.0543

Figure 7: The KBP 2016 Cold Start SF result of UNIST SAIL team

Sen Wu, and Ce Zhang. 2014a. Stanfords 2014 slot filling systems. *TAC KBP*, 695.

Gabor Angeli, Julie Tibshirani, Jean Wu, and Christopher D Manning. 2014b. Combining distant and partial supervision for relation extraction. In *EMNLP*, pages 1556–1567.

Gabor Angeli, Victor Zhong, Danqi Chen, Arun Chaganty, Jason Bolton, Melvin Johnson Premkumar, Panupong Pasupat, Sonal Gupta, and Christopher D Manning. 2015. Bootstrapped self training for knowledge base population. In *Proceedings of the Eighth Text Analysis Conference (TAC2015)*.

James Fan, Aditya Kalyanpur, David C Gondek, and David A Ferrucci. 2012. Automatic knowledge extraction from documents. *IBM Journal of Research and Development*, 56(3.4):5–1.

Thomas F Gordon. 2003. A use case analysis of legal knowledge-based systems. *Legal Knowledge and Information Systems (JURIX 2003), Frontiers in Artificial Intelligence and Applications*, pages 81–90.

Yifan He and Ralph Grishman. The nyu cold start system for tac 2015.

Bryan Kisiel, Bill McDowell, Matt Gardner, Ndapandula Nakashole, Emmanouil A Platanios, Abulhair Saparov, Shashank Srivastava, Derry Wijaya, and Tom Mitchell. Cmuml system for kbp 2015 cold start slot filling.

Farzaneh Mahdisoltani, Joanna Biega, and Fabian Suchanek. 2014. Yago3: A knowledge base from multilingual wikipedias. In *7th Biennial Conference on Innovative Data Systems Research*. CIDR Conference.

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.

Feng Niu, Ce Zhang, Christopher Ré, and Jude W Shavlik. 2012. Deepdive: Web-scale knowledge-base construction using statistical learning and inference. *VLDS*, 12:25–28.

Benjamin Roth, Nicholas Monath, David Belanger, Emma Strubell, Patrick Verga, and Andrew McCallum. 2015. Building knowledge bases with universal schema: Cold start and slot-filling approaches. *TAC2015*.

Lucas Sterckx, Thomas Demeester, Johannes Deleu, and Chris Develder. 2014. Using active learning and semantic clustering for noise reduction in distant supervision. In *4e Workshop on Automated Base Construction at NIPS2014 (AKBC-2014)*, pages 1–6.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 455–465. Association for Computational Linguistics.

Ngoc Thang Vu, Pankaj Gupta, Heike Adel, Hinrich Sch, et al. 2016. Bi-directional recurrent neural network with ranking loss for spoken language understanding. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6060–6064. IEEE.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *Journal of machine learning research*, 3(Feb):1083–1106.

Ce Zhang. 2015. *DeepDive: A data management system for automatic knowledge base construction*. Ph.D. thesis, Citeseer.