Natural language processing Term Project 2

Semantic Relations

1. Team member

湯忠憲	資料科學碩一	R06946003
劉宏國	資工碩一	R06922006
陳奎伯	資工碩一	P06922001

Agree to share your report with your classmates? (YES)

Code link: https://github.com/thtang/NLP/tree/master/project2

2. Methodology

We cast the task of determining a semantic relation and its direction as a classification task. Thus, to consider the relation and direction at the same time, we encode the ground truth to 19 classes including *other*. To begin with, we use RNN to learn the feature from word-embedded sentence vector directly. Several prior knowledges are introduced. For example, words between two entities play an important role to determine the relationship. Besides, we also feed the network with knowledge-based features. Finally, the technique of model ensembling is leveraged to achieve a robust performance. Details are shown as following sections.

2.1. LSTM architecture

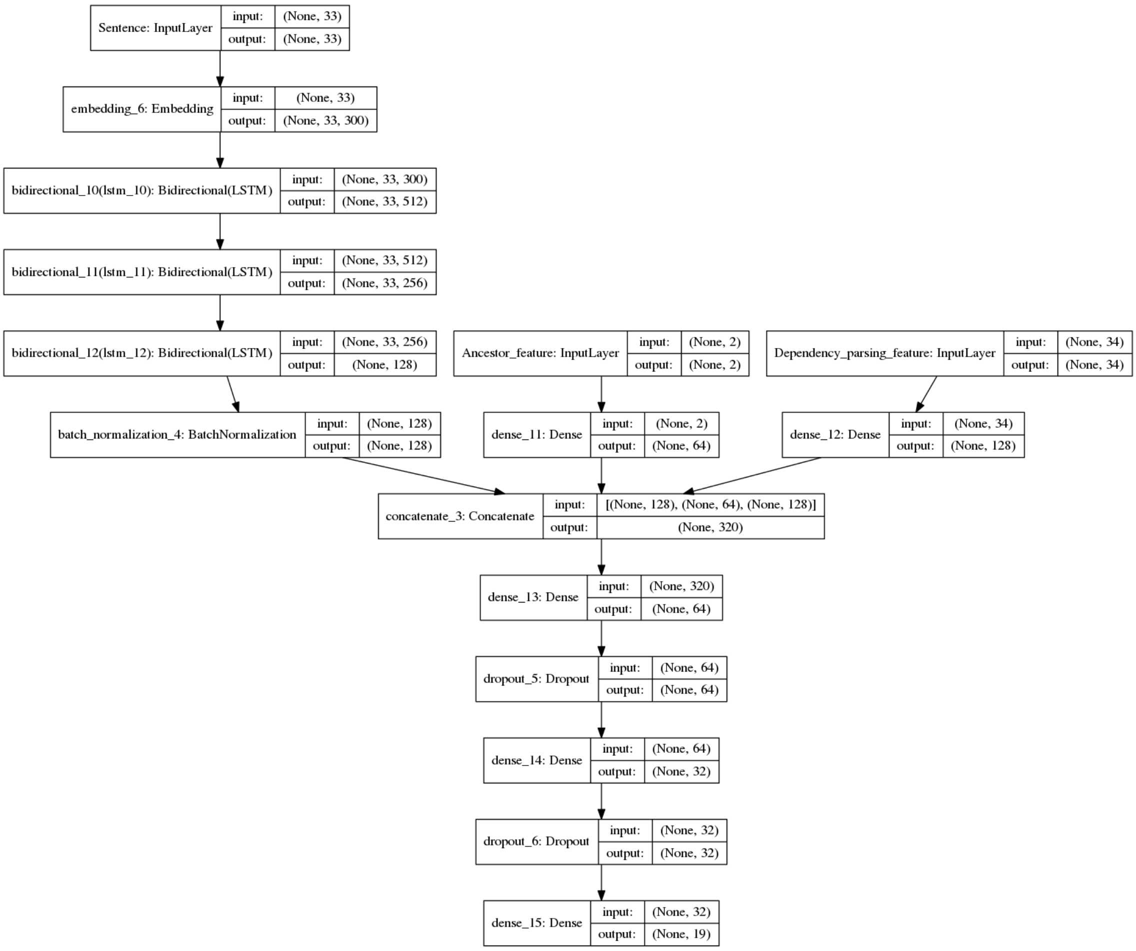

Fig. 1. Model architecture

Fig. 1 depicts the architecture of our classification model. Basically, there are two kinds of input channel. One is for sentences and the other type is for handcrafted features. Once the model extract features using LSTM and Dense layers respectively, we concatenate them and go through 3 Dense layers for predictions. Dropout, Bidirectional LSTM and BatchNormalization layer are adopted for better performance.

2.2. Sentence pruning

To meet the requirement of fixed input size for LSTM layer, each sentence would be padded with zero values. In our experiment setup, the max length is 87 tokens. However, based on our observation, most of the words have nothing to do with the entities relationship. Therefore, we prune the sentence so that limited context around the two entities are consider. An example is shown in Table 1. After processing this operation, the max length of training sentences drops, which leads to better performance for LSTM.

Table 1. Example of pruning sentences

Original sentence	*The fifty <e1>essays</e1> collected in this <e2>volume</e2> testify to most of the prominent themes from Professor Quispel's scholarly career.*
Pruned sentence	*The fifty <e1>essays</e1> collected in this <e2>volume</e2> testify to*

2.3. WordNet hypernym and PropBank hypernym

For WordNet approach, we use NLTK[1] WordNet to find entity’s synsets which have the same meaning as the entity first. Then, we get the hypernym words in all entity’s synsets as the features. An example is show in Table 2.

Table 2. Example of WordNet hypernym

Entity	*configuration and elements*
Hypernym features	*surroundings surround spatiality section portion part environs situation environment division substance…*

For PropBank approach, we use Google SLING[2] to parse our training sentence, it will return the structure of the PropBank format. An example in show in Table 3.

For list show in Table 3, ‘*bind.01*’ in column 1 means the frame which defines in PropBank website[3]; ‘*6*’ in column 3 means the token index in the sentence; ‘*ARG1_patient*’ in column 3 means the PropBank’s rules, it’s definition also show in the PropBank website; column 2 and column 5 are the token show in the sentence.

[1] https://www.nltk.org/

[2] https://github.com/google/sling

[3] http://verbs.colorado.edu/propbank/framesets-english-aliases/bind.html

Table 3. Example of PropBank structure

Sentence	*The child be carefully wrap and bind into the cradle by mean of a cord.*
PropBank structure	*['wrap.01', 'wrapped', 4, 'ARG1_patient', 'child'],* *['wrap.01', 'wrapped', 4, 'ARGM-MNR_manner', 'carefully'],* *['bind.01', 'bound', 6, 'ARG1_patient', 'child'],* *['bind.01', 'bound', 6, 'ARGM-MNR_manner', 'carefully'],* *['bind.01', 'bound', 6, 'ARG1_patient', 'cradle'],* *['bind.01', 'bound', 6, 'ARGM-MNR_manner', 'means']*

➤**words in the red font are entities**

Then, if verb's PropBank arguments (ex. *ARG1_patient* and *ARG1_patient'*) include two entities, we will get verb's hypernym words as our PropBank features. For the PropBank structure show in Table 3, verb *bound* include entity1 *child* and entity2 *cradle*, so we get *bound* hypernym words. An example shown in Table 4.

Table 4. Example of verb's hypernym

Verb	*bound*
Verb's Hypernym features	*stick attach bind fix relate confine bond restrain cover secure adhere fasten hold…*

2.4. Dependency parsing

Another method to acquire the relationship of two entities directly is to parse the dependency. In this task, we record whether *entity1* is ancestor of *entity2* and vice versa. For example, in sentence "The author of a keygen use a disassembler to look at the raw assembly code", "author" is the ancestor of "disassembler". This kind of cue could help us to classified the direction of relationship. The dependency tree is shown below.

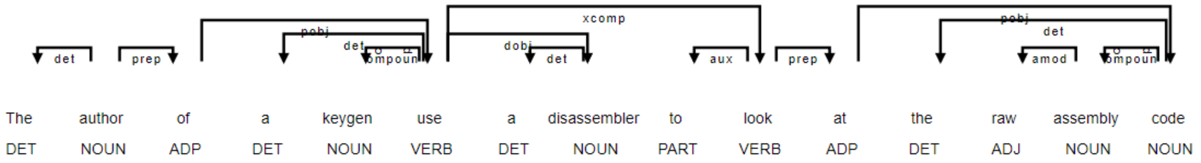

Fig. 2. Dependency tree

2.5. Ensembling

In order to make the classifier more powerful and robuster, we ensemble the models trained with different setting and features. Eventually, 4 models are used. The prediction probabilities for each class from the 4 models are averaged and select the most likely class as our predictive relationship.

3. Experiments Results

First of all, each word is represented by 300-dimensional word vector using GloVe pre-trained on Common Crawl (840B tokens, 2.2M vocab). Then a naive method is first implemented. We feed whole encoded sentence to the model mentioned in section 2.1 and the tensor go through the left LSTM track only. By doing so, a preliminary result with 70.02% F1-score is generated.

Secondly, we prune the input sentences then consider only entities and the words between given entities. The performance is greatly improved with 81.75% F1-score. Finally, we also try some handcrafted features as described in the previous section and feed them with one-hot encoding to extra channels in NN model. The F1-score reaches 82.10% using these features. With ensembling technique our approach goes beyond the method proposed by UTD, which is the first place of this task in 2010.

Table 5. F1 score of different setting

features	macro-averaged F1
Sentence only	70.02
Sentence from entity_1 to entity_2 (pruning)	81.75
+ WordNet hypernym and PropBank hypernym	81.77
+ Dependency parsing	82.10
Ensembling	**83.39**
UTD[4]	82.19

4. Discussion

In this term project, we build a system by using simple biLSTM model for recognizing semantic relations in SemEval-2010 Task 8. Beside, different kinds of features we add also can improve our f1-score. If we enhance our model become more deeper or more complicated, or add more background knowledge such as Verbnet[5], the f1-score may get more higher than before.

In addition, we provide another simple and effective approach: bag-of-words model, or BoW for short, is a way of extracting features from text for use in modeling. The model is only concerned with whether known words occur in the document, not where in the document.

It's worth to mention that we create a vocabulary of grouped words (n-grams that appear in the corpus) and allow the BoW model to capture a little bit more meaning from the document.

The following steps are helpful for reducing the amount of vocabulary:

- Ignoring case
- Ignoring punctuation
- Lemmatization
- Only take the part from entity_1 to entity_2 of the sentence

[4] http://www.aclweb.org/anthology/S10-1057

[5] http://verbs.colorado.edu/verb-index/index.php

Besides, we repeat prepositions: { of, in, into, on, onto, from, etc. } twice in the document to enhance relationship between objects. In a way this may be called Attention Mechanism.

The performance of the prediction is also measured by Macro-averaged F1: **75.12%**

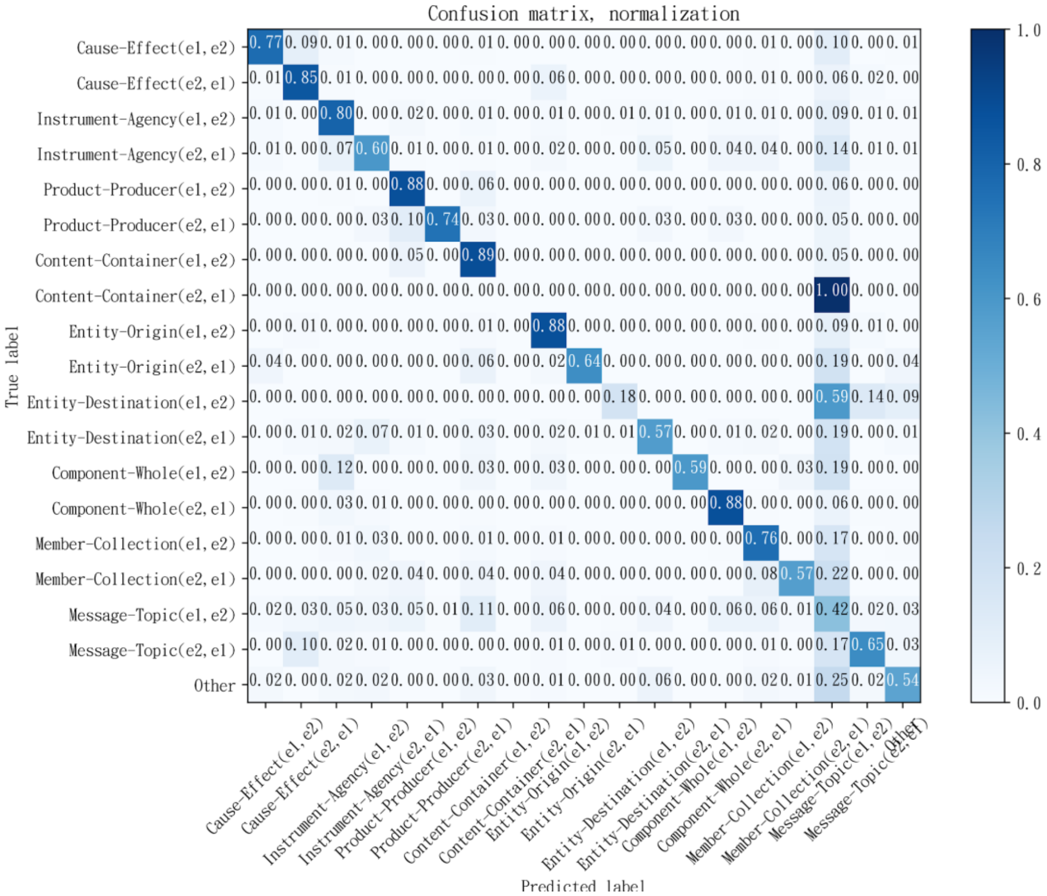

Fig 2. Confusion Matrix of BoW model's prediction