CSI5386 NLP Assignment II Text Entailment and Semantic Relatedness

Lingfeng Zhang 300134245 Ottawa University
Yu Sun 8472921 Carleton University

March 2020

GitHub clickable link:
`https://github.com/RichardChangCA/CSI-5386-Natural-Language-Processing-Assignments/tree/master/Assignment_2`
To fulfill this assignment, firstly we did some research together, like analysing the dataset distribution and basic model for these two tasks. After discussion, we began to do the implementation and wrote the report. For implementation, Lingfeng is mainly responsible for coding while Yu is for reviewing. For report, Yu is mainly for writing the report and Lingfeng is for revising. For classification task, the classes distribution is imbalanced. NEUTRAL is the majority class, poor classification model may be difficult to classify ENTAILMENT and CONTRADICTION, so precision, recall and F-measure are also used to evaluate models.

For the whole assignment, we used 3 different models to finish task 1 & 2: Bidirectional LSTM, Stacked Bi-LSTM, and bert. We finally found bert performs the best among these models, so we generate the file "Results.txt" by using trained bert model.

1 Task 1: Text entailment

Textual entailment tries to discern whether a continuous sentence pairs is related. There are three relationship categories: they are positive entailment when the first sentence proves the second sentence is true, negative entailment is on the contrary, neutral entailment means there are no correlation between the pairs.

For the text entailment, it contains two texts including T(the text) and H(the hypothesis) and three relations between them. The relations can be represented as ENTAILMENT, CONTRADICTION and NEUTRAL. What we need to do is to infer the relations from the text to the hypothesis. There are various approaches to solve this task, like Lexical Approaches, Machine Learning Approaches, Approaches based on Graph Matching and Semantics-based

Approaches[1]. In our assignment, we mainly focused on deep learning methods, using three different models including Seq2Seq with Bi-LSTM, Seq2Seq with stacked Bi-LSTM and Bert.

Sometimes, stop words may have a negative impact on the results of the classifier. So the dropout regularization method can be implemented to decrease the importance of stop words intelligently. In addition, dropout can also avoid overfitting to some extent.

Bi-directional LSTM because both premise and hypothesis can be reviewed in both forward and backward directions. It is suitable to find the relationship between premise and hypothesis.

For Seq2Seq with Bi-LSTM, we used the glove for word embedding, with three neuron nodes of binary value as the output, which is one-hot encoding method. The Seq2Seq with Bi-LSTM model is implemented by TensorFlow 1.x version. For Seq2Seq with stacked Bi-LSTM, it is similar with basic one, except using the stacked Bi-LSTM in hidden layer because the deeper hidden layers, the better performance in deep learning theoretically. The stacked Bi-LSTM model is implemented by TensorFlow 2.x version. For Bert, it is a end-to-end model and we used ktrain package for implementation and this package is based on TensorFlow 2.x version. The source code are listed in zip file, the three models are named by "text_entailment.py", "text_entailment_stacked_lstm.py" and "text_entailment_bert.py" respectively. In each python file, there are two main functions for model training and prediction, just run the file "text_entailment.py" under TensorFlow 1.x and other files under TensorFlow 2.x.

To evaluate the result against the gold standard, we use the following measurements: accuracy, consufion matrices, precision, recall and F-measure.

$$accuracy = \frac{TP + TN}{TP + FP + FN + TN}; \tag{1}$$

$$precision = \frac{TP}{TP + FP}; \tag{2}$$

$$recall = \frac{TP}{TP + FN}; \tag{3}$$

$$F - measure = 2. \frac{precision\ .\ recall}{precision + recall} \tag{4}$$

Since this task has 3 classes, one-vs-others method is used to build up confusion matrices. Firstly, we regard ENTAILMENT as one class and CONTRADICTION & NEUTRAL as another class, which converts a 3 classes task to a binary classes task. And so on.

Confusion matrix of Seq2Seq with Bi-LSTM, ENTAILMENT as positive, and others as negative.

Prediction outcome

actual value	p	n	total
p′	TP 0	FN 1414	1414
n′	FP 0	TN 3450	3450
total	0	4864	

precision:0.0
recall:0.0
f-measure:0.0

Confusion matrix of Seq2Seq with Bi-LSTM, NEUTRAL as positive, and others as negative.

Prediction outcome

actual value	p	n	total
p′	TP 2438	FN 292	2730
n′	FP 1628	TN 506	2134
total	4066	798	

precision:0.5996064928676832
recall:0.8930402930402931
f-measure:0.7174808711006475

Confusion matrix of Seq2Seq with Bi-LSTM, CONTRADICTION as positive, and others as negative.

Prediction outcome

actual value	p	n	total
p′	TP 462	FN 258	720
n′	FP 336	TN 3808	4144
total	798	4066	

precision:0.5789473684210527
recall:0.6416666666666667
f-measure:0.6086956521739131

Confusion matrix of Seq2Seq with stacked Bi-LSTM, ENTAILMENT as positive, and others as negative.

Prediction outcome

actual value	p	n	total
p′	TP 792	FN 622	1414
n′	FP 805	TN 2708	3513
total	1597	3330	

precision:0.49592986850344395
recall:0.5601131541725601
f-measure:0.5260710727333112

Confusion matrix of Seq2Seq with stacked Bi-LSTM, NEUTRAL as positive, and others as negative.

Prediction outcome

		p	**n**	**total**
actual value	**p′**	TP 1717	FN 1076	2793
	n′	FP 764	TN 1370	2134
	total	2481	2446	

precision:0.6920596533655784
recall:0.614751163623344
f-measure:0.6511186954872962

Confusion matrix of Seq2Seq with stacked Bi-LSTM, CONTRADICTION as positive, and others as negative.

Prediction outcome

		p	**n**	**total**
actual value	**p′**	TP 535	FN 185	720
	n′	FP 314	TN 3893	4207
	total	849	4078	

precision:0.6301531213191991
recall:0.7430555555555556
f-measure:0.6819630337794774

Confusion matrix of Bert, ENTAILMENT as positive, and others as negative.

Prediction outcome

actual value	p	n	total
p′	TP 1256	FN 158	1414
n′	FP 378	TN 3135	3513
total	1634	3293	

precision:0.7686658506731946
recall:0.8882602545968883
f-measure:0.8241469816272966
Confusion matrix of Bert, NEUTRAL as positive, and others as negative.

Prediction outcome

actual value	p	n	total
p′	TP 2363	FN 430	2793
n′	FP 230	TN 1904	2134
total	2593	2334	

precision:0.9112996529116854
recall:0.8460436806301468
f-measure:0.8774600816932789
Confusion matrix of Bert, CONTRADICTION as positive, and others as negative.

		Prediction outcome		
		p	**n**	**total**
actual value	**p′**	TP 616	FN 104	720
	n′	FP 84	TN 4123	4207
	total	700	4227	

precision:0.88

recall:0.8555555555555555

f-measure:0.8676056338028169

The result lists below, we can see Bert beats other two models, with 25% increment in the accuracy metric. Seq2Seq with Stacked Bi-LSTM get a better result than the basic one, but it is still worse than Bert.

We notice that Precision(Entailment) and Recall(Entailment) of Seq2Seq with Bi-LSTM are zero, probably because the data distribution is imbalanced and the Bi-LSTM model is not good enough.

	Seq2Seq with Bi-LSTM	Seq2Seq with Stacked Bi-LSTM	Bert
Accuracy	0.596	0.618	**0.860**
Precision(Entailment)	0.000	0.496	**0.769**
Recall(Entailment)	0.000	0.560	**0.888**
F-measure(Entailment)	0.000	0.526	**0.824**
Precision(Neutral)	0.600	0.693	**0.911**
Recall(Neutral)	**0.893**	0.615	0.846
F-measure(Neutral)	0.717	0.651	**0.877**
Precision(Contraction)	0.579	0.630	**0.880**
Recall(Contraction)	0.642	0.743	**0.856**
F-measure(Contraction)	0.609	0.682	**0.868**

2 Task 2: Semantic relatedness

For the semantic relatedness, we need to compute how much connection between the two concepts. Generally speaking, if two words are similar, they usually have high relatedness. But in another situation, although two words are different, they can also have high relatedness, like car and street. In this task, we use the similar model as task 1, except the difference of output layer, loss function and evaluation metrics. We replace the neuron node of continuous value for output layer. The source code are listed in zip file, the three models are named by "semantic_relatedness.py", "semantic_relatedness_stacked_lstm.py" and "semantic_relatedness_bert.py" respectively,just run the file "semantic_relatedness.py" under TensorFlow 1.x and other files under TensorFlow 2.x.

In this task, we convert a classification problem to a regression problem. In regression problem, we use "minimize MSE" as the loss function and the output of the model is a value rather than a class. After the whole prediction, predicted values are scaled range from 0 to 5.

To evaluate the result, we use the following measurement: mean squared error, Pearson correlation and Spearman correlation. We can see Bert have the best performance as well, But Seq2Seq with Stacked Bi-LSTM is worse that the basic one.

$$MSE = \sqrt{(\frac{1}{n})\sum_{i=1}^{n}(y_i - x_i)^2} \tag{5}$$

$$Pearson = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2(y_i - \overline{y})^2}}; \tag{6}$$

$$Spearman = 1 - \frac{6\sum d_i^2}{n(n^2-1)} \tag{7}$$

	Seq2Seq with Bi-LSTM	Seq2Seq with Stacked Bi-LSTM	Bert
mean squared error	0.717	1.014	**0.238**
Pearson correlation	0.588	0.176	**0.884**
Spearman correlation	0.496	0.170	**0.831**

3 Instruction of running source codes

Import dataset SICK_test.txt, SICK_test_annotated.txt, SICK_train.txt and SICK_trial.txt in the same folder with source codes.

Import glove pre-trained word-embedding files in the correct path.

Create folder named as: models, models_stacke_lstm, models_bert

In TensorFlow 1.x version:

python3 text_entailment.py

python3 semantic_relatedness.py
In TensorFlow 2.x version:
Installing ktrain package: pip3 install ktrain
python3 text_entailment_bert.py
python3 semantic_relatedness_bert.py
python3 text_entailment_stacked_lstm.py
python3 semantic_relatedness_stacked_lstm.py
All results can be generated as .txt files.
GitHub clickable link:
`https://github.com/RichardChangCA/CSI-5386-Natural-Language-Processing-Assignments/tree/master/Assignment_2`

References

[1] S. Ghuge and A. Bhattacharya, "Survey in textual entailment," *Center for Indian Language Technology, retrieved on April*, 2014.