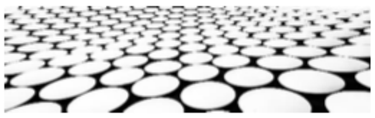
CSI5180 Topics in AI
Virtual Assistants



MODULE 3 – ASSIGNMENT

Building a paraphrase classifier

Report

CSI5180 - 2022 Winter Assignment 3

Building a paraphrase classifier

Student Kun Yan kyan008@uOttawa.ca (300259303)

Intro (2points)

- Describe the problem of binary classification of paraphrases

The goal of classification of paraphrases is to predict whether two sentences have the same meaning, and there are only two types of answers, that is, True (1) or False (0).

Dataset (6points)

- Describe what part of the dataset you will use

I used the full dataset SemEval-PIT2015 in my assignment, including 3 datasets (train.data, dev.data, and test.data). For each dataset, each line data is separated by "tab" and contains seven columns. To be more specific, they include 13063 rows, 4727 rows, and 972 rows, respectively.

Column1	Column2	Column3	Column4	Column5	Column6	Column7
Topic_Id	Topic_Name	Sent_1	Sent_2	Label	Sent_1_tag	Sent_2_tag

I did data preprocessing with Python pd.read_csv() to read them and output three separated csv files, so that I can input them into algorithm models later.

- Method for changing from graded evolutions to binary

For the train and dev datasets, the data of column "Label" contain six possible data pairs. I had mapped them into two types.

No.	Data pairs	Meaning (voting from Amazon Mechanical turkers)	Binary type

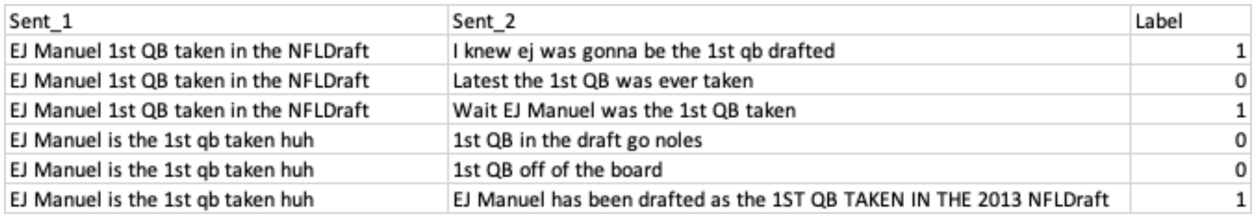
1	(0, 5)	0 is positive and 5 are negative	0 (non-paraphrases)
2	(1, 4)	1 is positive and 4 are negative	0 (non-paraphrases)
3	(2, 3)	2 is positive and 3 are negative	discard
4	(3, 2)	3 is positive and 2 are negative	1 (paraphrases)
5	(4, 1)	4 is positive and 1 are negative	1 (paraphrases)
6	(5, 0)	5 is positive and 0 are negative	1 (paraphrases)

For test dataset, the “Label” column is a format of a single digit from 0 to 5, so I have mapped 4 or 5 for paraphrases, 3 for discard, and 0 or 1 or 2 for non-paraphrases. The “Label” column is only compared with the predictive result to obtain the performance metrics.

As a result, the train dataset has 11529 rows, the dev dataset has 4141 rows, and the test dataset has 837 rows by deleting debatable rows and rows that include Nan value.

- Give examples of paraphrases and non-paraphrases (in a table)

For test dataset, the “Label” column is a format of a single digit from 0 to 5, so I have mapped 4 or 5 for paraphrases (1), 0 or 1 or 2 for non-paraphrases (0), and 3 for discard (delete the row).



Sent_1	Sent_2	Label
EJ Manuel 1st QB taken in the NFLDraft	I knew ej was gonna be the 1st qb drafted	1
EJ Manuel 1st QB taken in the NFLDraft	Latest the 1st QB was ever taken	0
EJ Manuel 1st QB taken in the NFLDraft	Wait EJ Manuel was the 1st QB taken	1
EJ Manuel is the 1st qb taken huh	1st QB in the draft go noles	0
EJ Manuel is the 1st qb taken huh	1st QB off of the board	0
EJ Manuel is the 1st qb taken huh	EJ Manuel has been drafted as the 1ST QB TAKEN IN THE 2013 NFLDraft	1

Methods (12points – 4points each)

- Describe the baseline algorithm and provide an example

I compared two sentences in the train dataset by calculating the Edit Distance between them as a baseline algorithm. Next, I input these distance data into Decision Tree Classifier algorithm to train the model. Then, I used the trained model to do prediction on the dev dataset.

- Describe method 1 algorithm, also provide an example

I improved the algorithm with Jaccard Distance between two sentences in the train dataset. Also, I input the improved distance data into the same Classifier algorithm with the first method to train the model. Then, I used the trained model to predict the paraphrases or not on the dev dataset.

- Describe method 2 algorithm, also provide an example

Furthermore, TFIDF method was used to analysis the meaning of two sentences.

Term Frequency-inverse document frequency (TF-idf): this looks at words that appear in both pieces of text, and scores them based on how often they appear. It is a useful tool if you expect the same words to appear in both pieces of text, but some words are more important that others.

It’s fast and works well when documents are large and/or have lots of overlap. It looks for exact matches, so at the very least you should use a lemmatizer to take care of the plurals.When comparing short documents with limited-term variety — such as search queries — there is a risk that you will miss semantic relationships where there isn’t an exact word match.

Results (6points)

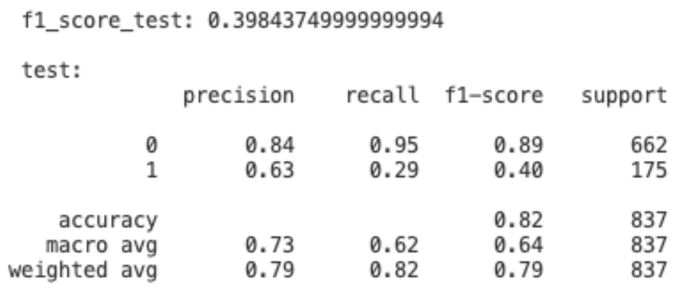
- Show comparative result of 3 methods on Dev

Algorithm	accuracy	F1-score	Precision (0)	Precision (1)	Recall (0)	Recall (1)
Edit Distance	0.6303	0.3629	0.68	0.47	0.81	0.30
Jaccard Distance	0.6928	0.4994	0.73	0.59	0.84	0.43
TFIDF	0.6829	0.5767	0.72	0.58	0.83	0.41

Obviously, the second method (Jaccard Distance) obtained the best performance metrics in accuracy, f1-score, precision, and recall.

- Show result of chosen method on Test

I chose the second method (Jaccard Distance) to do prediction on test dataset, and then compared the predictive result with the “Label” data in test dataset to calculate the accuracy (0.82), f1-score (0.89 for 0, 0.40 for 1), precision (0.84 for 0, 0.63 for 1), and recall (0.95 for 0 and 0.29 for 1).



```
  f1_score_test: 0.39843749999999994

  test:
               precision    recall  f1-score   support

            0       0.84      0.95      0.89       662
            1       0.63      0.29      0.40       175

     accuracy                           0.82       837
    macro avg       0.73      0.62      0.64       837
 weighted avg       0.79      0.82      0.79       837
```

- Discuss the results, including a qualitative analysis (showing a few examples that the best method got right or wrong)

In the result of prediction, I can observe that they are imbalanced datasets for train dataset and dev dataset, so the predictive metrics for paraphrases (1) are lower than non-paraphrases (0).

Conclusion (2points)

- Write a short conclusion about the experiment and future possibilities

In conclusion, it’s not simple to get a better predictive performance of paraphrases. Only analysing the words, their frequency and positions are not sufficient.

In the future, I am going to use more complex methods to do paraphrases prediction, like analysing POS and NE tags, using WordNet technology or n-gram models. I also want to use Machine Learning technique to deal with imbalanced datasets.

References (2points)

- Provide any reference you use for your programming

How to Rank Text Content by Semantic Similarity

- Provide a reference for the dataset

Data-Drive Approaches for Paraphrasing Across Language Variations