Text summarization
with location and intersection score

By
Jiranun Jiratrakanvong A20337992
Sihan Zhao A20351458

1 Introduction

Nowadays, there are several text summarization techniques, Extraction-based summarization, Abstraction-based summarization, and Aided summarization. In this project, the program has focused on Extraction-based summarization. Therefore, the program will not modify any sentences in the article.

The great amount of information produced by human on the Internet become more and more difficult to collect and recognize. As most of those information is in the format of human language except something like music or images, people should find a high efficient way to manage them. However, to deal with those language information, summarization is one of the most important phases of the whole natural language processing operations. Because sometime people need a more simplified or brief version of a set of information contain a large number of information.

For example, in my Apple Watch, there are many kinds of Apps of news such as Fox News, CNN News so on, but whatever App I choose, I found out it is very inconvenient to read news on an Apple Watch because the screen is too small to display enough words for even one paragraph. Moreover, although I could stand that and insist to read the whole news, I found it will be very desperate, because I have to flip almost hundred times of pages to finish a news story.

A good summarization of a document is just like a good name card or resume of a person when people decide if it deserved a more details to understand or interest, people first want to find a format of summarization. There are many reasons and situations where people need text summarization to give support to many other tasks just like the example of Apple Watch. The project we present here is to introduce a basic idea of text summarization of CNN’s news stories and show how is the algorithm work in the testing part.

2 Proposed solution

The basic idea is to give a program that can automatically give 5 top important sentences of an article that could be used to summarize a news story. The algorithm choose the sentences by giving each sentence a score of both we called location score and intersection score learned from training data.

2.1 Domain

In our project, we should firstly confirm which kind of information on the internet is the input. Finally, we choose cnn.com to be the only source of training and test data of English news stories. The reason why we chose cnn.com instead of some website such as Wikipedia is that we think the news storied of cnn.com is in a very good organized format which is easy to for extraction and summarization. If we choose to give text summarization without any restrict of domain, we may face some very difficult problem such as the difference of language. By the way, as we are going to collect the train and test data by ourselves to tag by hand, we thought the news story from a famous media may be easier to read and understand. So we were planning to collect at least 50 sets of data, each set contains a page of the whole news story and a page of top 5 important sentences which together could summarize the news story tagged by our own hand.

2.2 Format transformation

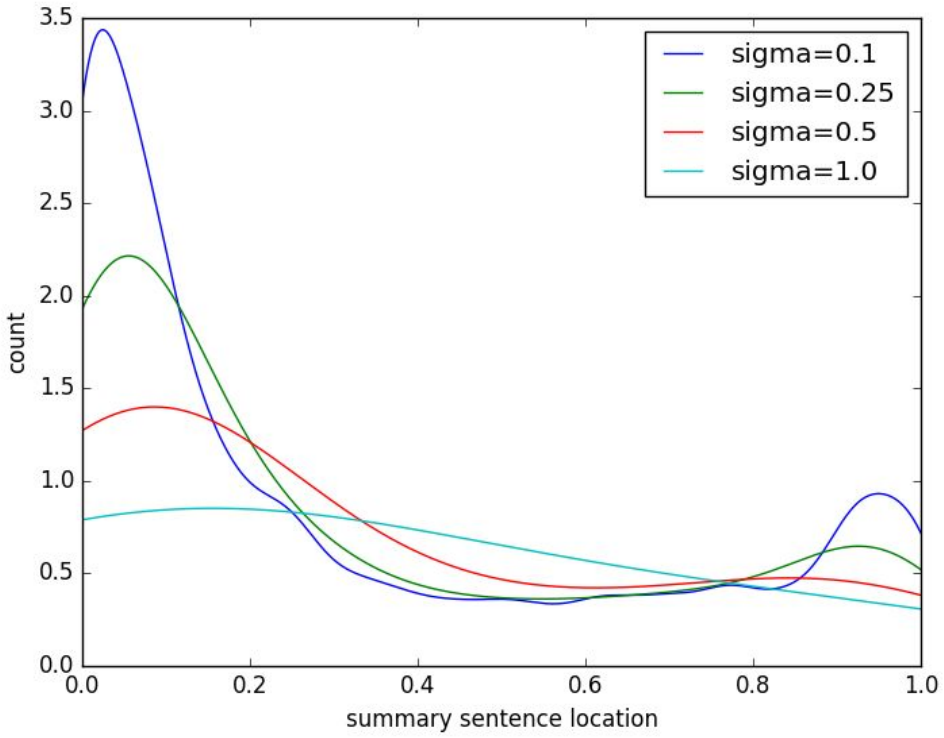
In the news articles we choose to use for training. Before whatever the algorithm or idea of summarization, we should first clear the sentences format of the text. In a news story, there are many content such as dash or colon, we treat all kind of these content as a period which means we define the format of every sentences so that we could do the future job of training and testing. This should the be the first step of our algorithm and programming.

2.3 location and intersection scores

The program will find the N best sentences from the article by choosing sentences with the highest score. To make sure the reliability of the algorithm, we use two ways of score calculation. One is named location score, the other is named as intersection score.

The location score means that if the location of a sentence is more likely to be a summary sentence of an article, it will get a higher score than others. The likelihood of a location of an article is more like a summary sentence is calculated by training data. For example, in most of news stories in cnn.com, the first sentence is always a summary sentence. So in our algorithm, sentence in the beginning tains to get higher score than

other sentences at the beginning. To calculate the location score, the location of each sentence will be matched with a density graph produced by training data. However, there is a parameter that we need to specify, sigma. Sigma is used to produce estimated gaussian graph from training data. Its value will directly affect the size of the graph. See Image below to see how sigma affect the density graph.



Then, we also use another score named intersection score. For an intersection score, the score of each sentence depends on its words. A sentence's score will be higher if the words in that sentences are mentioned a lot of times in the article.

To combine these two kinds of scores, we use two parameters to calculate a total score for each sentence. S function means the total score of a sentence. 'I' means the i^{th} sentence of an article. A very important job for us is to find the best pair of location and intersection ratio to define the most efficient way of the score calculation through the cross-validation with different pair of parameters.

$$S(i)=loc\_ratio*loc\_score(i)+int\_ratio*int\_score(i)$$

3 Implementation detail

Finding best parameters

In evaluation.py, best location ratio, intersection ratio, and sigma will be generated by accuracy calculation of all possible value. Because they are continuous values, they were cut into range. Location ratio possible values are 0.0, 0.05, 0.1, … , 1.0. Intersection ratio = 1.0 - Location ratio, and sigma possible values are 0.0, 0.1, 0.2, …, 1.0. As a result, there are 20x10 = 200 possible cases. In each case, the accuracy will be calculated. Finally, the best parameters will be chosen by the highest accuracy.

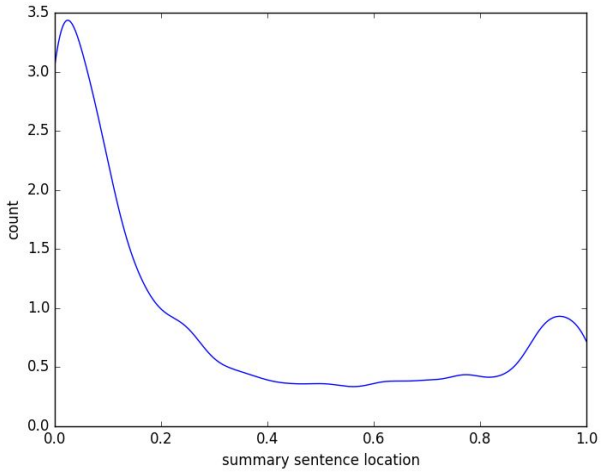
For the accuracy calculation, the program will calculate how similar between the summary gotten from the program and the summary from manual label data. For example, if the manual summary sentences are [1,2,3,5] while the summary sentences from the program are [1,2,4,6], the accuracy is intersection of those sets divide by union of those sets. In this case,

$$\text{accuracy} = \frac{|[1,2,3,5] \cap [1,2,4,6]|}{|[1,2,3,5] \cup [1,2,4,6]|} = \frac{2}{6} \approx 0.333$$

4 Results and Evaluation

Cross-validation has been used in the evaluation part. There are 5 folds for cross-validation with 100 pairs of article and summary. Therefore, each fold contains 80% of training data and 20% of testing data. The accuracy is the average accuracy of every fold.

After running evaluation.py, the best parameters have been generated (location ratio = 0.1, intersection ratio = 0.9, and sigma = 0.1. While sigma = 0.1, the density graph became as below.



With these parameters, the average accuracy is approximately 40% which satisfy our expectation.

5 References

1.Rada, M.and Paul, T. 2004. *TextRank: Bringing Order into Texts,* Department of Computer Science University of North Texas

2.Sebastian T., Rishabh I., Hoachen W. and Jeff B. 2014. *Learning Mixtures of Submodular Functions for Image Collection Summarization.* In Advances of Neural Information Processing Systems (NIPS)

3.Ramakrishna B., Rishabh I., Ganesh R. and Jeff B. 2015. *Summarizing Multi-Document Topic Hierarchies using Submodular Mixtures*. Annual Meeting of the Association for Computational Linguistics (ACL)

4.Kai W., Rishabh I., and Jeff B. 2015. *Submodularity in Data Subset Selection and Active Learning.* Proc. International Conference on Machine Learning (ICML)