ALGORITHM-ORIENTED TEXT SUMMARIZATION OF FINANCIAL NEWS

Yibo Feng

Contents

1. Introduction

With more and more text data generated by the Internet, the problem of text information overload is becoming more and more serious. Text summaries are designed to transform text or text collections into short summaries containing key information. According to the output type, it can be divided into extractive abstract and generative abstract. Extractive abstract extracts keywords from the source document to form an abstract. Generative abstracts generate new words and phrases based on the original text to form an abstract.
Broadly speaking, there are two ways to summarize the text:

- Extraction based summarization
  In an abstract based on extraction, a lot of the most important words are extracted from the text and combined to form an abstract. In machine learning, extracting a summary usually involves weighing the basic parts of a sentence and using the results to generate a summary, but the result may be grammatically inaccurate. There are a lot of various types of methods and algorithms to measure the weight of sentences, then rank them based on their similarity and then connect them to create a summary.
- Abstraction-based summarization
  In abstract-based summary, deep learning techniques are intended to interpret and shorten the source text. Since abstract machine learning algorithms can generate new phrases that represent the most important information in the original text, this can help overcome the grammatical errors of extraction summarization.
  Although the abstract form performs better, the development of its algorithm requires complex machine learning techniques and complex language modeling. To produce reasonable output, this method must solve various problems. Therefore, the method of extracting text summaries is still widely popular.

Our project uses the extractive text summarization technique based on two algorithms: TF-IDF and TextRank. We want to explore what kind of algorithm is suitable for the summarization of news. So we implemented the two algorithms and compared their results.

2. Data Description

2.1 Dataset Description

Our dataset is the US Financial News Articles. We got them from the Kaggle. It includes one hundred seventy-nine thousand news from January to March in two thousand eighteen.

2.2 Data Preprocessing

For preprocessing the text, we start with the sentence tokenize and lower case all the words. Then remove irrelevant and redundant information that may not provide more value to the meaning of the text. The steps of data cleaning include remove punctuation, number, and special characters and remove stop words.

3. Algorithms

3.1 Reasons for choosing TFIDF and TextRank

Based on article "Text Summarization Techniques: A Brief Survey", text summarization got attraction in the 1950s. There was a significant research which explained a method to extract sentences from text or documents using features like word or phrase frequency. TFIDF is such an algorithm mainly using word frequency to measure sentences' weights and extract sentences. The reason we choose TextRank because the core of it is to build similar matrix. Similarity of two sentences in similar matrix is calculated by computing cosine similarity with TFIDF weights for words. It could conclude that two algorithms have connections but run with different tracks. Therefore, our goal using them is to see how different result they could bring.

3.2 TFIDF

The TFIDF is a numeric measure that calculates weight of a word is to a sentence or a document in the text collection. There are several steps to achieve the goal. To be specific, firstly, preprocessing consists of the operation needed to enhance feature extraction including tokenization, removing stop words, lower-case words, word stemming. Second, it's feature extraction. It is used to extract features of the documents by obtaining the sentences in text based on its importance which mainly means word frequency and given the value from zero to one. Then, sentence selection is the sequence which uses descending order to sort sentences. The highest rank will be considered as summary. Finally, summary generation is to put sentences into summary in the order of the position in original documents. Figure 1 shows the process and diagram for extractive text summarization.

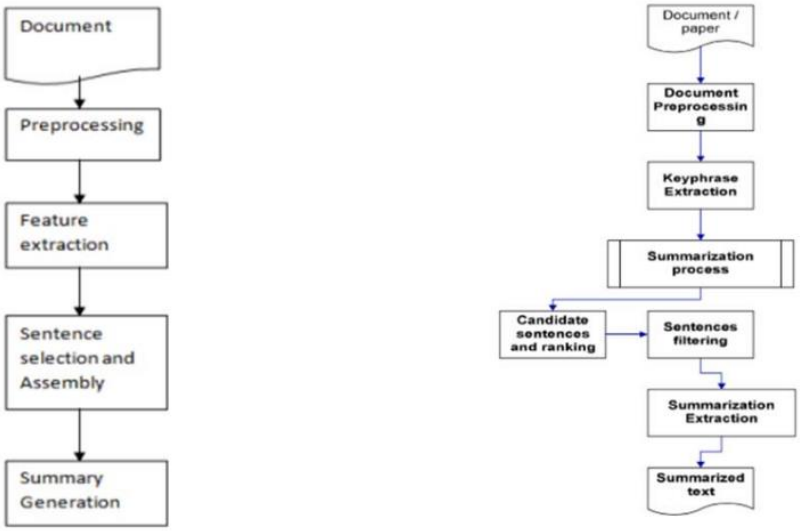

Figure 1 Text Summarization Extraction Process and Diagram

3.3 TextRank

TextRank is associated with Google's PageRank which is used for ranking webpages for online research results. Before understanding TextRank, it is necessary to know logic and intuition behind it. Basically, PageRank assumes that the rank of a webpage W depends on

the importance of a webpage suggested by other webpages in terms of links to the pages. i.e if a webpage X has a link to webpage W, “X” contributes the importance to “W”. Basically, it is used this kind of logic to build similar matrix with multiple webpages. Figure 2 shows the matrix between “X”, “Y”, “W” and “Z”.

	Page W	Page X	Page Y	Page Z
Page W	0	0.2	0.25	0.33
Page X	-	-	-	-
Page Y	-	-	-	-
Page Z	-	-	-	-

Figure 2 Webpage Similar Matrix

TextRank uses same logic here with PageRank but with some subtle changes. First, sentences will in place of webpages. Similarity matrix is filled with similarity score between sentences which usually is cosine similarity. Just give an example. “He is a nice guy. He has a lot of friends. Raj is his best friend.” Figure 3 shows similarity matrix between these sentences.

	He is a nice guy	He has a lot of friends	Raj is his best friend
He is a nice guy	0	0.53	0.2
He has a lot of friends	0.53	0	0.9
Raj is his best friend	0.2	0.9	0

Figure 3 Sentence Similar Matrix

The process for TextRank includes several steps. First, preprocessing is necessary such as sentence tokenization, removing stop words. Second, find word embedding such as Word2Vec() to calculate each word score of text and combine them as sentence embedding. Then, use sentence embedding to build similarity matrix and convert it to a network/graph. Final step is to select specific number of sentences to build summary.

4. Implementation

4.1 TFIDF

TFIDF score equals to TF score * IDF score. we calculated TFIDF score for each word and it is basic idea to measure the frequency of sentence. It cost about 40 mins to run the code. The data set we used was financial news articles from 2018-1 to 2018-3. Figure 4 shows brief process for TFIDF algorithm.

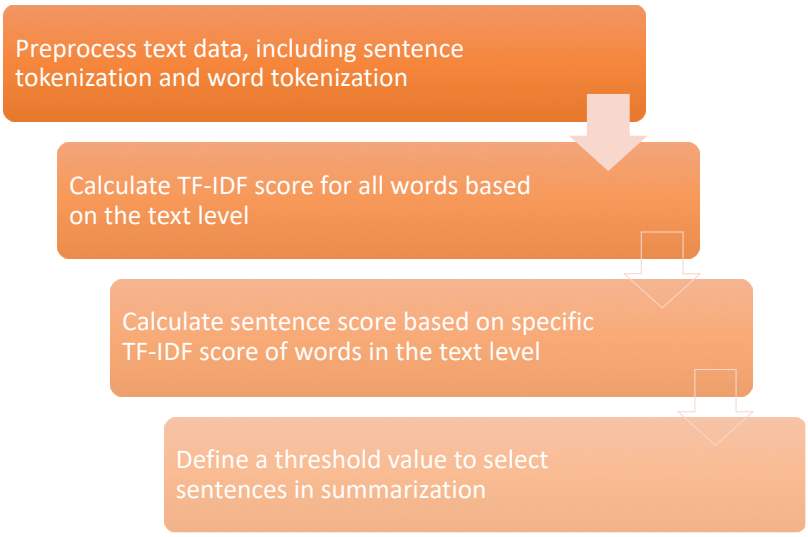

Figure 4 Text Summarization Process in TFIDF

We initially read into data frame. We created frequency matrix of the words in each sentence. Second, we calculated term frequency and generated a matrix. The term frequency was defined as the count for each word divided counts for all words. Third, we used term frequency matrix to create a table for documents per word which means "how many sentences contain a word". Fourth, building a IDF matrix and its formula is log based e with (Total documents /total sentences contains with term). Here the document is a line of text and term is a word. Next step was to build a TFIDF matrix by combining above two matrixes. TFIDF algorithm is made of two algorithms multiplied together. Scoring the sentences in each line of text is differs with different algorithms. Here, we were using TFIDF score of each word in a sentence to give weight to the text. The final step was to find the threshold and extract sentences to create text summary. There may have different ways to calculate a threshold value. We were calculating the average sentence score. Then, we selected all sentences for a summarization if the sentence score was more than average score. Figure 5 shows a random example from the result set. It was clearly that the summarization did not show many key words as text in title such "North Korea", "World-Level Tourist Project" and "Olympics".

```
In [155]: df['title'][666]
Out[155]: 'N.Korea trumpets world-level tourist project ahead of Olympics'

In [153]: tfidf['summarization'][666]
Out[153]: "['Experts say that tourism is an important part of Kim's plans to boost the North Korean economy.', 'It is one of a shrinking range of North Korean cash sources not
specifically targeted by international sanctions.', 'Last year, America banned its citizens from visiting North Korea and there are no up-to-date statistics on current visitors.',
'China said more than 237,000 Chinese visited in 2012 but it stopped publishing the statistics in 2013.', 'For comparison, eight million Chinese visited South Korea in 2016.',
'(Reporting by Josh Smith; Editing by Nick Macfie)']"
```

Figure 5 Example Summarization for TFIDF

4.2 TextRank

The preprocessing was similar with TFIDF algorithm, but running the code was a time-consuming job which took about 5.5 hours. Figure 6 shows progress from raw data to summarization using TextRank.

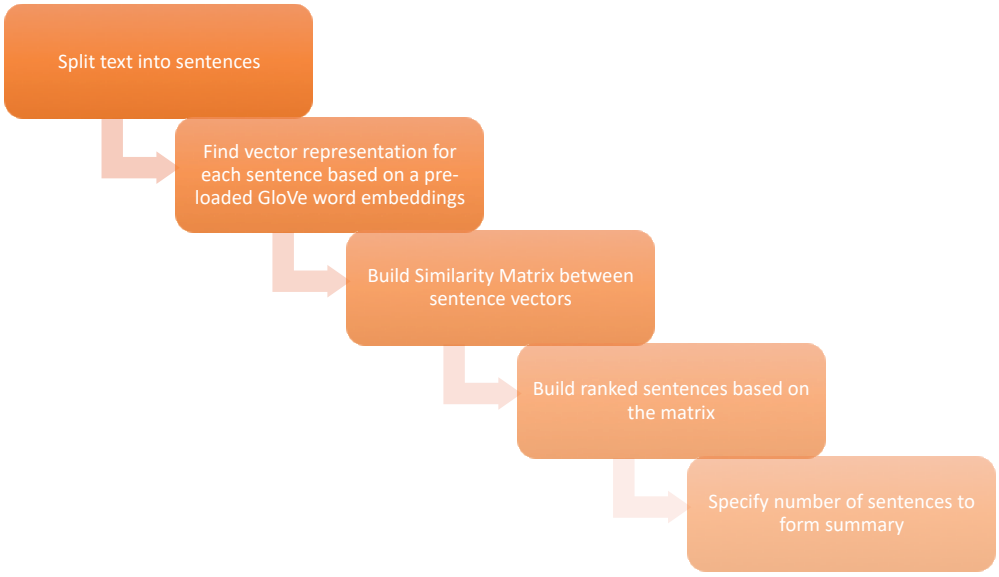

Figure 6 Text Summarization Process for TextRank

We firstly broke the text into individual sentence by sentence tokenization. We then downloaded Glove as vector representation of words. These embeddings will be used to create sentences vectors. We could use bag-of-words to create word features for sentences in text. However, it ignores order of words and the result of the feature will be huge. This word embeddings set was from the pre-trained Wikipedia 2014 Glove vectors. We extracted word embeddings for about 4000 different terms in the dictionary. The next step was to create sentences vector in each line of text. The core of algorithm was finding similarities between the sentences and calculated cosine similarity for pair of sentences. Building ranked sentences based on matrix was using PageRank algorithm. Then, defining a threshold number to specify the number of sentences to form the summary. At the end, it is time to use the threshold number to choose sentences based on descending order rankings for summary. Figure 7 shows the same example extracted from the result. Compared with previous one, this summarization was slightly different that some key words came in appear in the first sentence such as “North Korea”, “World-Level Tourist Project”.

```
In [155]: df['title'][666]
Out[155]: 'N.Korea trumpets world-level tourist project ahead of Olympics'
In [154]: df['textrank_summarization'][666]
Out[154]: 'SEOUL, Jan 25 (Reuters) - North Korea advertised a new "world-level" tourist project in coastal Kangwon province on Thursday, a statement that coincides with plans by
Winter Olympics host South Korea to participate in joint sporting and cultural events in the area.Already facing criticism for plans to march under a unity flag and field a combined
Korean ice hockey team, the administration of South Korean President Moon Jae-in may come under further pressure if it is seen to be endorsing Kim's luxury getaway on North Korea's
east coast.In April last year, for example, Kim used the beach near Wonsan's new airport to unleash an artillery drill described by state media as the country's largest ever.'
```

Figure 7 Example Summarization for TextRank

5. Evaluation

5.1 Methods Review

Evaluation methods of text summarization is an essential part of the development of text summarization. However, artificial evaluation not only costs too much money but wastes a lot of time with strong subjectivity. Seminars about automatic abstracts have appeared at

different famous conferences, such as DUC and TSC. This provides a standard summary training and evaluation platform for researchers. At the same time, it promotes the development of automatic summary technology.

According to Josef's research, he listed several evaluation methods in different situations with their pros and cons. [5](Steinberger, 2012) Text quality evaluation cannot be automatically, thus we didn't take it into consideration in our analysis. Extrinsic evaluation focuses on the comparison among different summarization systems. However, it is task-based, which aims at a specific task without analyzing sentences in the summarization tasks. Therefore, our analysis mainly considers content evaluation, which directly evaluates the content of summarization. As Josef summarized, co-selection can match identical sentences, but it ignores the fact that two sentences written differently could still contain the same information. Meanwhile, content-based measures calculate the similarity of sentences in reference texts and candidate texts. There are some basic methods, such as BLEU, METEOR and ROUGE. BLEU is an evaluation measure of the accuracy of a model with multiple correct outputs, which are used in evaluating text translation. METEOR is an evaluation method based on weighted harmonic mean of single precision and single word recall rate. Unfortunately, it is mainly used in machine translation.[6](Banerjee, 2005) Another mainstream evaluation metric is ROUGE, which is mostly used for text summarization.

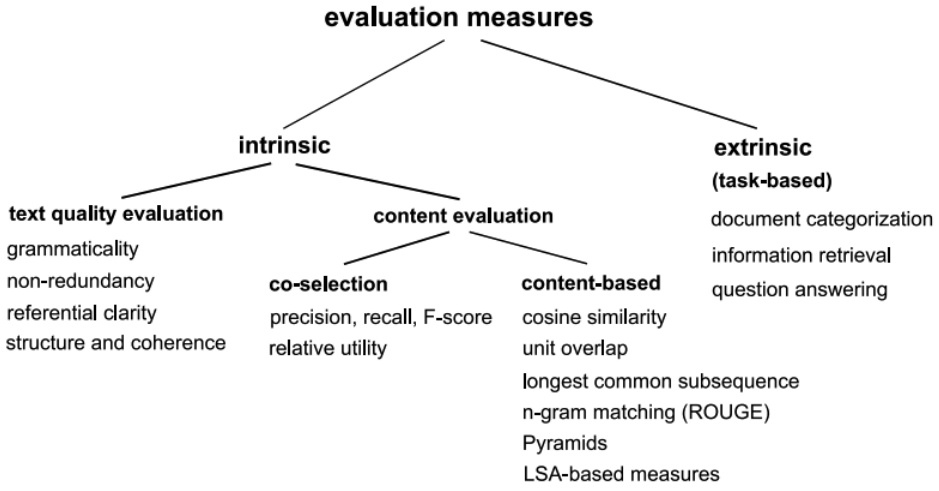

Figure 8 The taxonomy of Text Summarization in Josef's Paper

5.2 ROUGE

ROUGE is a recall-oriented method of evaluating text summarization. It was proposed by Chin-Yew Lin (2004)[7]. The main idea of this method is to compare the automatic summary with artificial standard summary by counting the overlapped units' number to evaluate the quality of summary. ROUGE has several types depending on different features used to calculate recall, including ROUGE-N and ROUGE-L.

- ROUGE-N
  ROUGE-N calculates scores according to the number of grams, which supposes that the appearance of N^{th} words is only related to the former N-1 words. For example, ROUGE-1

counts recall by matching unigrams while ROUGE-2 counts recall by matching bigrams. According to the formula, ROUGE-N counts the total number of n-grams in all reference abstracts and find how many of them appear in the candidate summary. $Count_{match}(gram_n)$ means the number of n-gram which appears in both reference summary and candidate summary. The denominator is all numbers of n-grams in reference summary.

$$\text{ROUGE-}n = \frac{\sum_{C \in RSS} \sum_{gram_n \in C} \text{Count}_{match}(gram_n)}{\sum_{C \in RSS} \sum_{gram_n \in C} \text{Count}(gram_n)},$$

- ROUGE-L
  ROUGE-L is based on longest common subsequence (LCS). LCS works on making the sequence with the maximum length of the common subsequence as the longest common subsequence of both based on two given sequences. We applied LCS into summary-level in our analysis. According to the formula, LCS compares each sentence r_i in reference summary with all sentences in candidate summary, and results in union LCS as the matching of sentence summary r_i.

$$R_{lcs} = \frac{\sum_{i=1}^{u} LCS_{\cup}(r_i, C)}{m}$$

$$P_{lcs} = \frac{\sum_{i=1}^{u} LCS_{\cup}(r_i, C)}{n}$$

$$F_{lcs} = \frac{(1+\beta^2) R_{lcs} P_{lcs}}{R_{lcs} + \beta^2 P_{lcs}}$$

5.3 Implementation

Because of the particularity of our dataset, we used the title of each news as the reference summary. Although titles couldn't reveal all ideas in articles, they still generalize the main ideas of articles.

	title	text
0	EMERGING MARKETS-Mexican peso seesaws over dol...	(Updates prices, adds Trump comments) By Rodri...
1	Migrants must visit Nazi concentration camps, ...	BERLIN (Reuters) - New migrants to Germany mus...
2	Euro zone businesses start 2018 on decade high	Euro zone businesses start 2018 on decade high...
3	Russia's Lavrov says 'unilateral actions' by U...	MOSCOW (Reuters) - "Unilateral actions" by the...
4	Lawmakers to Justice Department: Keep online g...	ATLANTIC CITY, N.J. (AP) - Federal lawmakers w...
...	...	...
4995	BRIEF-China First Capital Group Posts FY Loss ...	March 29 (Reuters) - China First Capital Group...
4996	BRIEF-Smart-Core Holdings Says FY Net Profit A...	March 26 (Reuters) - Smart-Core Holdings Ltd:\...
4997	Confidence in crude market fundamentals 'seems...	Confidence in crude market fundamentals 'seems...
4998	Maersk Line says four missing after container ...	(Reuters) - Maersk Line, the world's biggest c...
4999	BRIEF-Deutsche Konsum REIT Acquires Retail Pro...	March 5 (Reuters) - DEUTSCHE KONSUM REIT AG:\n...

Figure 9 Sample Title and Text

We called Rouge() functions to calculate scores of recall, precision and F-score for each text. Since F-score is to evaluate the similarity between reference summary and candidate summary, we use it as evaluation for each text. We only considered ROUGE-1, ROUGE-2 and ROUGE-L when calculating scores, because the value of ROUGE-N will be very small when N is larger than 3. Then, we calculated the average scores as the result of the application of each algorithm.

```
In [7]: np.mean(evaluation_fscore_rl)
Out[7]: 0.08973725926897239

In [8]: np.mean(evaluation_fscore_r1)
Out[8]: 0.08246057990824136

In [9]: np.mean(evaluation_fscore_r2)
Out[9]: 0.04258869067362772
```

Figure 10 F-score of TF-IDF

```
In [15]: np.mean(tr_evaluation_fscore_rl)
Out[15]: 0.14921761585128573

In [16]: np.mean(tr_evaluation_fscore_r1)
Out[16]: 0.12746742126949015

In [17]: np.mean(tr_evaluation_fscore_r2)
Out[17]: 0.08040700479453841
```

Figure 11 F-score of TextRank

6. Results Analysis

Compared the summarization of news whose title is “North Korea trumpets world-level tourist project ahead of Olympics” extracted by TFIDF and TextRank. The summarization we get from TFIDF mainly said tourism is important to boost the North korea economy, and there are fewer tourism from America and China visited the North korea in recent years. But the summarization we get from TextRank indicates the North korea would develop a “world-level” tourist project with Winter Olympics host South Korea. It’s obvious that the text from TextRank is closer to the news title and it’s better to summarize the news article.

Compared two algorithms, the mainly different is that Textrank is based on sentence vectors to filter sentences, and the runtime is 5 hours to extract the summarization, while TFIDF mainly focus on the TF-IDF score calculation of all words based on the text level, it runs 40 minutes to summarize. Textrank has a higher F score and a better performance than TF-IDF, which shows in Figure 12.

```
In [143]: print('The average F-Measure score for TFIDF model
is',np.mean(evaluation_f))
The average F-Measure score for TFIDF model is 0.08973725926897239

In [144]: print('The average F-Measure score for TextRank model
is',np.mean(tr_evaluation_f))
The average F-Measure score for TextRank model is 0.14921761585128573
```

Figure 12 Performance Comparison

7. Future Work

For the future work, some of our results are not good to summarize news content. Some results looks like Figure 13, the summarization didn't work well. It only extracts some periods and it's not a normal text. So it hopes to be optimized in the future. The F-score is very low for both two algorithms , because title can't fully explain the content of news article. When doing evaluation, manually extracted abstracts may perform better as a reference to evaluate the results of algorithms. And the methods of evaluation need to be improved. Rouge-L only calculates a longest subsequence, and the final value ignores the influence of other candidate of longest subsequence and shorter subsequence. Finally, the runtime of TextRank needs to be reduced.

```
  File "/opt/anaconda3/lib/python3.7/site-packages/rouge/
rouge_score.py", line 253, in rouge_n
    raise ValueError("Hypothesis is empty.")

ValueError: Hypothesis is empty.


In [103]: tr_df['textrank_summarization'][42613]
Out[103]: '...'

In [104]: tr_df['title'][42613]
Out[104]: 'Pepper...and Salt - WSJ'
```

Figure 13 Bad Performance of Summarization

8. Conclusion

Text summarization is a common problem in natural language processing. In this project, we preprocess dataset by sentence tokenization, removing punctuation as well as number and special characters, using all lowercase and removing stopwords. Then we choose TextRank and TFIDF to extract summariazation from news content, and use ROUGH to evaluate two models' performance. Compared the results of two model, the TextRank has a better performance than TF-IDF.

Reference

[1] Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., D., E., B., J., & Kochut, K. (2017). Text Summarization Techniques: A Brief Survey. International Journal of Advanced Computer Science and Applications, 8(10). doi:10.14569/ijacsa.2017.081052

[2] Christian, H., Agus, M. P., & Suhartono, D. (2016). Single Document Automatic Text Summarization using Term Frequency-Inverse Document Frequency (TF-IDF). ComTech: Computer, Mathematics and Engineering Applications, 7(4), 285. doi:10.21512/comtech.v7i4.3746

[3] Joshi, P. (2020, July 19). Automatic Text Summarization Using TextRank Algorithm. Retrieved November 30, 2020, from https://www.analyticsvidhya.com/blog/2018/11/introduction-text-summarization-textrank-python/

[4] Panchal, A. (2020, November 29). Text Summarization using TF-IDF. Retrieved November 30, 2020, from https://towardsdatascience.com/text-summarization-using-tf-idf-e64a0644ace3

[5] Steinberger, J., & Ježek, K. (2012). Evaluation measures for text summarization. Computing and Informatics, 28(2), 251-275.

[6] Banerjee, S., & Lavie, A. (2005, June). METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization (pp. 65-72).

[7] Lin, C. Y. (2004, July). Rouge: A package for automatic evaluation of summaries. In Text summarization branches out (pp. 74-81).