Extractive Text Summarization using Latent Semantic Analysis

Runhai Lin

New York University

rl3192@nyu.edu

Abstract

Text summarization is a task to condense a large piece of text to a shorter but concise and accurate summarizations which focus on the main topic. One section in text summarization, extractive text summarization is to find out the most topic-related sentence among all sentences. One algorithm of extractive text summarization is latent semantic analysis (LSA). In this paper, I applied for LSA, built a system for text summarization in two approaches and evaluated their ROUGE-N performance in English.

1. Introduction

As the growth of the amount of texts on Internet, text summarization has become an increasingly popular realm in natural language processing. Extractive text summarization, to generalize summary of texts based on the sentences in the text, is one of the major realm in text summarization. The history of extractive text summarization derived from 1950s which was to focus on the surface level of analysis and developed rapidly in 21st century.

The paper here focuses on an algorithm called latent semantic analysis. The contribution to use LSA in the realm of text summarization was made by Yihong Gong and Xin Liu in 2002. Using similar way in latent semantic indexing, they ran SVD to discover the hidden topic among a large texts.

In this paper, the author will describe the application of text summarization using LSA. The rest of paper consists of four sections. Section 2 describes previous research work in LSA. Section 3 describes the procedure of text summarizations and presents algorithm applied. Section 4 shows the collection of dataset. Section 5 presents the performance of summary and analyzes it. Section 6 concludes the paper.

2. Previous Work

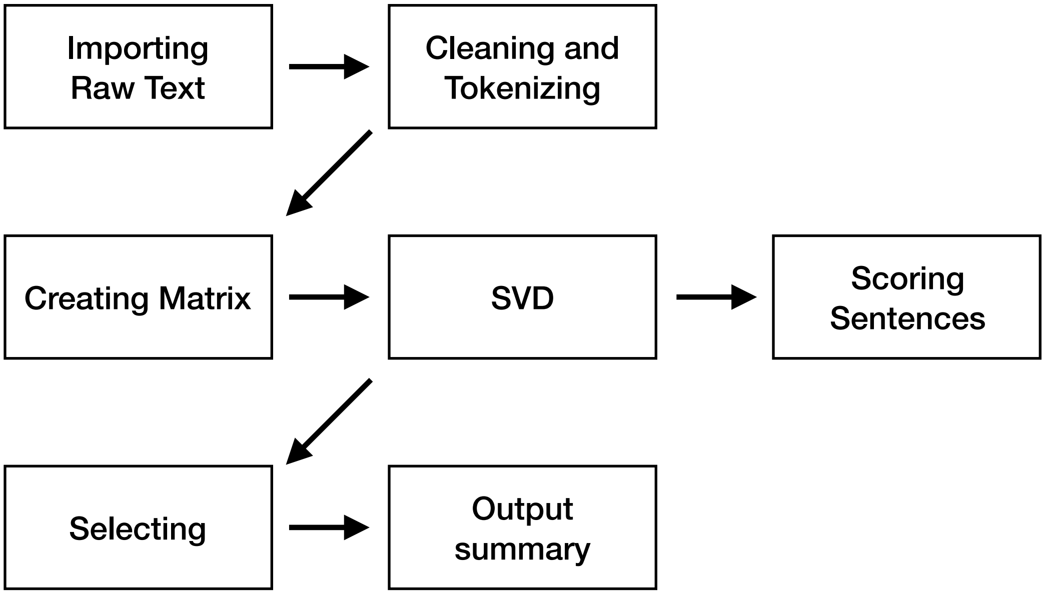
Yihong Gong and Xin Liu present the idea of sentence matrix constructions. Their paper proposes an idea to split the large documents into sentences, construct vector for each sentence based on term frequency and merge them into a matrix.

Yihong Gong and Xin Liu also present two generic ways to analyze the sentence matrix. One of them is summarization by LSA. They propose this SVD-based summarization method with instruction to decompose matrix and select the top k-th sentence in the right singular matrix.

Thomas Hofmann provides a method based on mixture decomposition derived from a latent class model. The aspect model constructed associates class variable with observations. For model fitting, Thomas Hofmann adapts EM algorithms to fit the model.

Josef Steinberger and Karel Ježek improve the LSA algorithm. After the matrix has been decomposed into three parts, they propose to compute length as score for each sentence. This sentence scoring method is able to evaluate the sentences even the number of dimension of reduced space n is unknown. In addition, it overcomes the disadvantage of Gong and Liu sentence selection algorithm when large (but not largest) index value interferes the results.

3. Procedure and Algorithm



In this section, there will be description about how the text summarization system works.

3.1 Input Data Cleaning

In this part of the text summarization system, input raw texts are formatted and cleaned to prepare for sentence matrix construction. The input raw texts are split into two sets: document texts and manual summaries. The documents texts are tokenized by sentences. Manual summaries are stored and grouped by index.

Because LSA is an algorithm without external information, unnecessary terms like stop-words and punctuations are removed. In addition, a dictionary including all terms appear in the documents is created.

3.2 LSA algorithm

3.2.1 Sentence matrix creation

The first step of LSA is to turn cleaned texts into a matrix A which computer can understand and analyze. Every column vector of the matrix A_i represents the term-frequency of words in sentence i. If a document has m terms and n sentence, the text summarization system will construct a m × n matrix.

There are multiple choices to fill out the cell values. In this system, TFIDF values of the term fills out the matrix entry. TFIDF refers to the multiplication of term frequency and inverse document frequency. The formula is in the following.

$$TFIDF = TermFrequency * log(\frac{NumberofSentence}{NumberofSentenceContainingT})$$

The formula shows that if a term is more frequent in a sentence, the TFIDF score for this term is high. It means the sentence is more related with this term. If the number of sentences containing T is high, then TFIDF is low because this term becomes less necessary to distinguish a specific sentence.

An example sentence matrix is like the following:

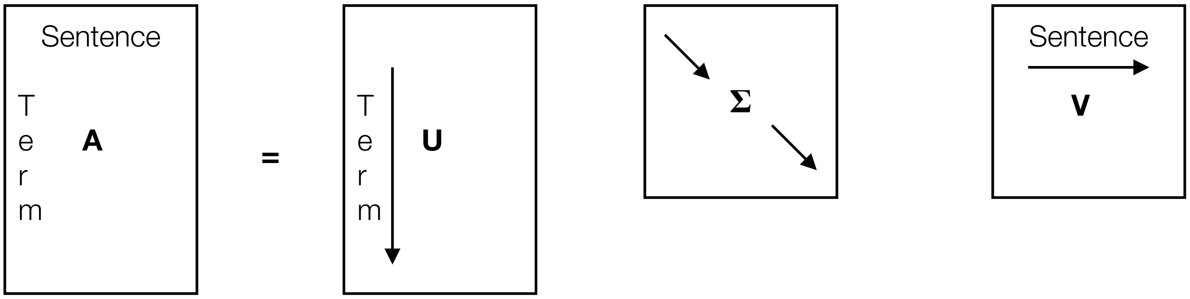
	Top	Seed	Never	Around	…
“top seeds roger…”	1.146…	1.447…	0	0	
never mind the '	1.146…	0	1.447…	1.690…	
…					

3.2.2 Singular Value Decomposition

Singular value decomposition is a method to factorize a matrix and turn it to be the composition of normal square matrix. For a $m \times n$ matrix A, a standard value decomposition can be applied that

$$A = U\Sigma V^t$$

U is a $m \times n$ left singular matrix, Σ is a $n \times n$ diagonal matrix, V is a $n \times n$ right singular matrix. Since singular value decomposition is not unique, it is possible to build Σ with descending diagonal value.



If the rank of matrix A is r, the SVD decomposes it into r linearly independent base vector. From transformation view, SVD derive a mapping between m dimensional and r dimension vector space. V^T represents weighted term-frequency matrix.

From the semantic view, SVD presents the hidden relationship between sentences and related topics. Related terms are more likely to appear close to each other. After transformation, terms with similar pattern will have close cell value because all sentence vectors are transformed and projected in r-dimensional topic spaces. Therefore, sentences following a certain pattern will be projected close to other who have similar one.

3.2.3 Sentence Selection

Based on the result from SVD, there are different algorithms to select the top n topic-related sentence in sentence matrix.

In this system, two algorithms are selected and compared.

Algorithm of Gong and Liu: A higher cell value represents closer relations between a topic and a sentence. This algorithm picks the top k important topics and it picks the sentence with highest absolute cell value.

Algorithm of Steinberger and Jezek: Steinberger and Jezek points out some disadvantages of Gong and Liu's algorithm. First, the number of sentences chosen is the same as the number of reduced dimensions, which may include some unnecessary sentences. Second, sentences with high score may be emitted if only one sentence is picked in that topic. So Steinberger and Jezek's algorithm is to compute the length of sentence vectors and select the top k vectors with largest length. Their algorithm gives an overall evaluation of sentence matrix.

4. Data set collection

The document sets trained, developed and tested in this paper is stories from CNN-DailyMail dataset, which was created by Hermann et al. in 2015. This dataset contains 90266 online news articles from CNN. The average number tokens in one news article is 781. Every news article is followed by multi-sentence summaries in average 3.75 sentences or 56 tokens.

The dataset is split into three corpora. The training corpus takes up 80% of documents, the developing corpus takes up 15%, and the training corpus takes up 5%.

5. Evaluation and Analysis

Evaluation of text summaries is complex topic in natural language processing. In this paper, ROUGE scores are used for evaluations. The ROUGE scores measure n-gram co-occurrence between computer generated results and human manual summarizations. In order to better evaluate how many key terms are included in the summarization, ROUGE-1 and ROUGE-2 are particularly applied.

Since the average number of sentences of human manual summary is 3.75, in this evaluation, reduced dimension is set to be 2 and 3.

Two summarizers are evaluated:

1 Gong and Liu LSA Summarizer

2 Josef Steinberger and Karel Ježek Summarizer

Table1: ROUGE-1 scores with dimension = 2

	Precision	Recall	F-score
Gong and Liu	0.298	0.170	0.217
Steinberger and Jezek	0.312	0.172	0.222

Table2: ROUGE-2 scores with dimension = 2

	Precision	Recall	F-score
Gong and Liu	0.02	0.017	0.019
Steinberger and Jezek	0.022	0.017	0.020

Table3: ROUGE-1 scores with dimension = 3

	Precision	Recall	F-score
Gong and Liu	0.352	0.156	0.216
Steinberger and Jezek	0.367	0.180	0.242

Table4: ROUGE-2 scores with dimension = 3

	Precision	Recall	F-score
Gong and Liu	0.02	0.012	0.015
Steinberger and Jezek	0.021	**0.017**	0.018

The result of the evaluation shows that the difference between two LSA summarizers. The Steinberger and Jezek algorithm performs better in recall in these four evaluations. The better performance in recall indicates that it is possible for the Steinberger and Jezek algorithm to remove the disadvantage of limited sentence selection in a certain topic.

However, the difference between two algorithms is small as reduced sentence k is small. Based on the current evaluation, it is hard to say that Steinberger and Jezek algorithm performs better in excluding unnecessary sentences.

Another noticeable results are LSA algorithm works well in extracting a certain key term as topic. However, the performance for bigram is pretty low.

6. Conclusion

This paper briefly discusses the LSA algorithm and its principle. It shows out how this algorithm can be used in the field of text summarizations.

The ROUGE-score from evaluation provides insights for better performance in Steinberger and Jezek algorithms but difference is small. In the future, I plan to build better system with better ROUGE-n performance. My goal is to construct a better system which can not only find related sentence but also generate more human-readable summary.

7. Reference

[1] Gong ,Yihong & Liu, Xin. “Generic text summarization using relevance measure and latent semantic analysis.” In Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '01). Association for Computing Machinery, New York, NY, USA, 19–25. 2001. DOI:https://doi.org/10.1145/383952.383955

[2] Xu, Wei & Liu, Xin & Gong, Yihong. “Document clustering based on non-negative matrix factorization.” In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval (SIGIR '03). Association for Computing Machinery, New York, NY, USA, 267–273. 2003. DOI:https://doi.org/10.1145/860435.860485

[3] Hofmann, Thomas. “Probabilistic Latent Semantic Analysis.” *UAI* (1999).

[4] Ozsoy, Makbule & Alpaslan, Ferda & Cicekli, Ilyas. “Text summarization using Latent Semantic Analysis.” J. Information Science. 37. 405-417. 10.1177/0165551511408848. (2011).

[5] Steinberger, Josef & Jezek, Karel. “Using Latent Semantic Analysis in Text Summarization and Summary Evaluation.” (2004).

[6] Landauer, Thomas & Foltz, Peter & Laham, Darrell. “An Introduction to Latent Semantic Analysis." Discourse Processes. 25. 259-284. 10.1080/01638539809545028. (1998).

[7] Chen, Danqi & Bolton, Jason & Manning, Christopher. “A Thorough Examination of the CNN/Daily Mail Reading Comprehension Task.” 2358-2367. 10.18653/v1/P16-1223. (2016).

[8] http://nlpprogress.com/english/summarization.html

[9] https://github.com/abisee/cnn-dailymail