Summarization of Online Document Repositories

Dr. M. Hanumanthappa[1] B R Prakash[2] Rashmi S[3]

[1]*Professor, Department of Computer Science & Applications, Bangalore University, Bangalore.*
[2,3] *Research Scholar, Department of Computer Science & Applications, Bangalore University, Bangalore.*
hanu6572@hotmail.com, brp.tmk@gmail.com, rashmi.karthik123@gmail.com

Abstract

As websites on the Internet in the Web 2.0 era have become more interactive,there has been an explosion of new user-generated content. The goal of Summarization Pipeline for Online Repositories of Knowledge(SPORK) is to be able to identify important key topics presentedin multi-document texts, such as online comment threads. While most otherautomatic summarization systems simply focus on finding the top sentences representedin the text, SPORK separates the text into clusters, and identifies differenttopics and opinions presented in the text. SPORK has shown results ofmanaging to identify 72% of key topics present in any discussion and up to 80%of key topics in a well-structured discussion.

Key words: Summarization, Natural language processing, Pre-processing, Term Frequency-Inverse Document Frequency.

1. Introduction

Natural language processing has been used extensively on the Internet andhas been put to a variety of tasks to mitigate this problem. Keyword extraction can be used to better tag certain documents,like research articles. all based in part on natural language processing, which help users sift through the enormous amount of content to help findrelevant matches. Search engines rely on the use of natural language processingto return more relevant results.[1] They use various techniques for information retrievaland for getting a deeper semantic understanding of what search queriesmean. Text extraction and summarization can be used to display summaries of search results is widely used in question and answering systems.

While many natural language processing techniques can benefit users, automatictext summarization is the key to mitigating information overload producedby these sources. A summary can be loosely defined as a textthat is produced from one or more text(s), that conveys importantinformation inthe original text(s), and that is no longer than half of the original text and usuallysignificantly less." The idea is that automatic summarization should reducethe amount of text in a document or multiple documents while still retainingall of the important information. There are several different types of text summarizationall suited to solve slightly different problems[2]. A detailed analysis ofcurrent summarization techniques and a novel approach will be discussed.

2. Related work

Automatic text summarization is an active and developing field, and one ofthe important applications within natural language processing. Within automatictext summarization, several techniques are used to accomplish a variety oftasks. Generally speaking, there are two distinctive types of text summarization:extractive and abstractive. Extractive summarization approaches create a summaryby using sentences or phrases from the corpus text. They focus on choosingrelevant and salient sentences that can be used to represent the document. On theother hand, abstractive summarization approaches create a novel summary froma corpus text. This type of technique can rely on creating a semantic representationof the text and uses natural language generation to create novel sentencesthat do not necessarily show up in the corpus text. In general, most techniques that are not defined as an extractive method are still considered abstractive orpartially abstractive approaches. Recently, abstractive summarization techniqueshave begun to see some research and development.

2.1 Extractive Summarization

2.1.1 Supervised Summarization

Automatic text summarization methods can use a supervised, unsupervised,or semi-supervised approach[3]. A supervised or semi-supervised approach requiresmanually labelled data to train on. In addition, when training on labelled data,all data comes from one domain. This means that running the summarization willonly work for that specific domain, minimizing the portability of the approach.

2.1.2 Multi-Document Summarization

In addition to taking an unsupervised approach, web blog comment summarizationstands out in the sense that multiple texts are being summarized. Eachcomment in the thread acts as its own text and each text can contain different viewpoints or ideas. This requires a slightly modified approach for extractingsentences in order to properly reduce redundancy while still gathering uniqueinformation about separate topics[4]. A modified approach could also address the difficulty of multiple topics.

2.1.3 Query-Based Summarization

Next, we look at generic- versus query-focused summarization. Within textsummarization, methods can either target their summaries using specific queriesor topics or remain completely general. Several studies more have focused on summarizing text based on certain topics using a querybasedapproachp5].

2.2 Abstractive Summarization

2.2.1 Validation Approaches

The nature of text summarization requires the summary to be judged by ahuman, since computers cannot yet correctly compute the quality of summarizations.This means that for a number of summarizations that my researchproduces, there will also have to be accompanying human-made summarizations.However, the amount of human work involved in the process can be minimized.The solution to this involves two metrics that have been created to analyze thesetypes of problems: ROUGE and BLEU[6,7].

3. Design Overview

The typical pipeline for an extractive text summarization approach beginsby taking a corpus and tokenizing the text into sentences. This relates to bothsingle and multi-document corpora. Once the sentences are tokenized, they canoptionally be processed to optimize for different types of extraction analysis.

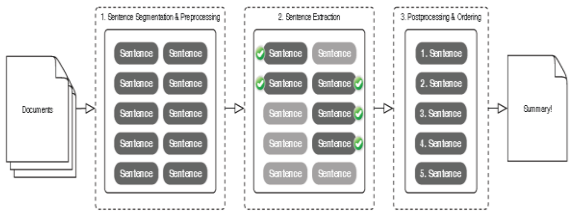

Figure 1: Basic Summarization Pipeline. This figure shows thesimplified pipeline of a basic extractive text summarization approach.

A typical processing technique, sentence simplification, can include removingpunctuation or stop-words. With clean, tokenized sentences, the next step is toextract salient sentences that can be used in the summary. We used a number ofmethods to extract relevant sentences .The last step of the typicalsummary process involves ordering and post-processing the extracted sentences.This includes making sure that topically coherent sentences are put together,and that punctuation is correct. The end result generally depends on intendedformat of the summary, whether it be for summarizing a product review, oractually making a coherent human-readable paragraph of text. Either way, ingeneral, it is assumed to be a readable text that should adequately summarizethe corpus.

As shown by the Figure 1, the steps are divided into three general blocks:pre-processing, extraction, and post processing. While this is a simplified overviewof a general summarization approach, it illustrates the major steps required formost tasks[8].

Unlike traditional summarization methods, which usually just attempt to capturethe most important sentences within the document, the approach designedin this work attempts to capture a summary based on different topics presenting Reddit comment thread. This method tries to avoid the problems of redundancypresent in other methods, and attempts to highlight the different ideaswithin the document[9]. The work outlined in this research focuses on the first twomajor steps of Figure 1, sentence segmentation & pre-processing and sentenceextraction, and aims to extract opinions located within different topics of thetext. These steps are further broken up into sub-steps and are explained in moredetail.

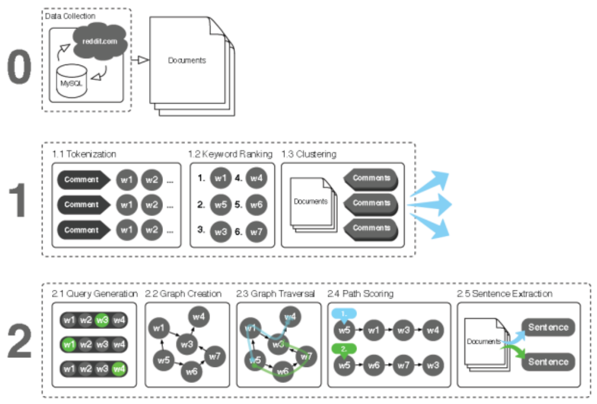

Figure 2: SPORK Summarization Pipeline suggested in this work. It focuses on the

first two steps of the basic extraction method in Figure 1

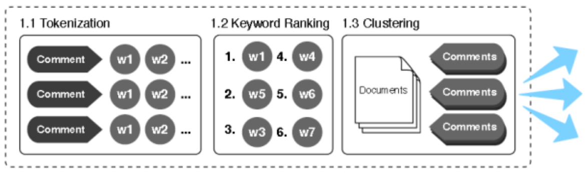

Figure 3: Pipeline: Pre-processingof the summarization pipeline.

4. Preprocessing

The collected data is fed directly to the actual summarization process. Thesummarization process begins with some pre-processing required to help get amore consistent summarization. This part of the pipeline is illustrated in Figure3, and contains a couple of steps.

The pre-processing proceeds as follows:
First, comments are parsed and tokenized for use in the pipeline. Thisusually encompasses tokenizing the comments into sentences and furthermoretokenizing the sentences into a list of words. Both of these tasks canbe done using several different techniques, but the NLTK recommendedtokenizes are used. Additionally, part-of-speech tags for the tokenized sentencesare required later in the pipelineSecond, a keyword ranking approach is implemented to help find the importanceof certain phrases and connections usedThird, once keywords are ranked, the comments are clustered together into several different topics. This step helps streamline the graph creation andthe traversing process by reducing complexity as well as maintaining morerelevant graphs.

4.1 Tokenization and Tagging

The seemingly trivial task of separating sentences from a chunk of text, actuallyhas its complexities. A simple sentence tokenizer would just separate sentences every time a terminating punctuation is encountered. The complexityoccurs because periods do not have to terminate a sentence in the English language.Periods can be used for abbreviations, and, therefore, can be placed inthe middle or at the end of a sentence. The NLTK recommends sentence tokenizeris the PunkSentenceTokenizer[10].

Finally, we use the Stanford POS Tagger to accurately tag the tokenized text.POS Tagging is essential to numerous natural language processing tasks. Difficultiesin POS tagging can be attributed to the ambiguities of the English language[11]. Not only do words have different part-of-speech tags in different contexts, butsometimes words can have multiple meanings.

4.2 Keyword Ranking

The next step in the pipeline is ranking keywords (Figure 2). Thisstep requires tokenized sentences and words and aims to rank the importanceof all keywords in the comment thread. This is done by obtaining the TermFrequency-Inverse Document Frequency (TF-IDF) rank for each word. TF-IDFdefines the relationship between the numbers of times a word shows up withina specific document to the inverse of the number of other documents the wordshows up in the corpus[12].

$$tfidf(w,d,D) = f_{w,d} * \log\left(\frac{|D|}{f_{w,D}}\right)$$

4.3 Clustering

When using the graph-based summarization approach it is beneficial to haveredundant overlapping opinions. This is tough since the Reddit comment thread scontain so many different discussions and opinions. In order to improve the efficiency of the summarization approach we want to group certain comments in agiven comment thread together. This focuses the discussion into more manageableand cohesive groups. We achieve this by clustering like-comments together.

5. Conclusion

The main goals of automatic text summarization can be accomplishedusing extractive techniques, abstractive summarization is usually referred to whentalking about a \true" summarization. Just as humans would read, parse, andgenerate a novel summary of some text, abstractive summarization generates newsentences not found in the original text. This type of summarization is highly desired and also extremely difficult. These difficulties come from a combinationof creating a deep semantic understanding of the text and using natural languagegeneration. An attempt at doing this is shown in the Opinosis Graph.Although this is considered a \shallow" abstractive approach, it still generatesnovel sentences not found in the original text. These sentences are based onredundancies between words found in the graph. This wasa major hurdle in this research and motivated the need for sentence extractionapproaches. Although the Opinosis Graph was tested on a much more limitedcorpus (manually labelled redundant reviews), it proved not to transfer over wellto the more unstructured text used in this work. This leaves many possibilitiesfor future work, including exploring different approaches in validating correctsentence paths. This would eliminate the need for the sentence extraction techniques and transform this pipeline into abstractive, albeit shallowly abstractive,solution.

10. References

[1] P. Achananuparp, X. Hu, and X. Shen.The evaluation of sentence similaritymeasures.In Data Warehousing and Knowledge Discovery, pages 305{316.Springer, 2008.
[2] S. Bird, E. Klein, and E. Loper.Natural Language Processing with Python.O'Reilly Media, 2009.
[3] A. Celikyilmaz and D. Hakkani-T□ur.Discovery of topically coherent sentencesfor extractive summarization. In Proceedings of the 49th AnnualMeeting of the Association for Computational Linguistics: Human LanguageTechnologies - Volume 1, HLT '11, pages 491{499, Stroudsburg, PA, USA,2011. Association for Computational Linguistics.
[4] D. Das and A. F. Martins. A survey on automatic text summarization.Literature Survey for the Language and Statistics II course at CMU, 4:192{195, 2007.
[5] K. Filippova. Multi-sentence compression: finding shortest paths in wordgraphs. In Proceedings of the 23rd International Conference on Computational Linguistics, COLING '10, pages 322{330, Stroudsburg, PA, USA,2010. Association for Computational Linguistics.
[6] K. Ganesan, C. Zhai, and J. Han.Opinosis: a graph-based approach to abstractivesummarization of highly redundant opinions. In Proceedings of the23rd International Conference on Computational Linguistics, COLING '10,pages 340{348, Stroudsburg, PA, USA, 2010. Association for ComputationalLinguistics.
[7] T. S. N. L. P. Group. Stanford pos tagger faq.http://nlp.stanford.edu/software/pos-tagger-faq.shtml.
[8] C. Ho, M. A. A. Murad, R. A. Kadir, and S. C. Doraisamy. Word sensedisambiguation-based sentence similarity. In Proceedings of the 23rd Inter-national Conference on Computational Linguistics: Posters, pages 418{426.Association for Computational Linguistics, 2010.
[9] F. Liu and Y. Liu. From extractive to abstractive meeting summaries: canit be done by sentence compression? In Proceedings of the ACL-IJCNLP2009 Conference Short Papers, ACLShort '09, pages 261{264, Stroudsburg,PA, USA, 2009. Association for Computational Linguistics.
[10] Y. Lu, C. Zhai, and N. Sundaresan.Rated aspect summarization of shortcomments. In Proceedings of the 18th international conference on Worldwide web, WWW '09, pages 131{140, New York, NY, USA, 2009. ACM.
[11] K. Ganesan, C. Zhai, and J. Han.Opinosis: a graph-based approach to abstractivesummarization of highly redundant opinions. In Proceedings of the23rd International Conference on Computational Linguistics, COLING '10,pages 340{348, Stroudsburg, PA, USA, 2010.Association for ComputationalLinguistics.
[12] C. Ho, M. A. A. Murad, R. A. Kadir, and S. C. Doraisamy. Word sensedisambiguation-based sentence similarity. In Proceedings of the 23rd International Conference on Computational Linguistics: Posters, pages 418{426.Association for Computational Linguistics, 2010.