IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 3 Issue 6, June 2016
ISSN (Online) 2348 – 7968 | Impact Factor (2015) - 4.332
www.ijiset.com

Pruning Document Data using Top-K Rule for Effectively Summarized Text

Ms. Priya J Patel[1], Prof. Pravin G Kulurkar[2]

[1] Department of Computer Science & Engg, VIT, RTMNU University, Nagpur, Maharashtra, India

[2] Department of Computer Science & Engg, VIT, RTMNU University, Nagpur, Maharashtra, India

Abstract

Automatic text summarization is a process to reduce the volume of text documents using computer programs to create a text summary with keeping the key terms of the documents. Due to cumulative growth of information and data, automatic text summarization technique needs to be applied in various domains. Text Summarization was showed to be an improvement over manually summarizing the large data. It summarizes the salient features from the text by preserving the content and serves the meaningful summary. To design an algorithm that can summarize a document by extracting key text and attempting to modify this extraction using a thesaurus and to reduce a given body of text to a fraction of its size, maintaining coherence and semantics. This summarization method can be done in natural language processing approach integrated with rule mining.

Keywords: *Automatic Summarization, Extraction, Natural Language Processing, Top-K Rule*

1. Introduction

Nowadays with increase of information, users need to have access to effective methods in order to search for the requested information. In most cases, people study the summary of a document rather than the whole. Automatic text summarization is a solution for this issue. Automatic summarization is the process of condensing textual content into a concise form for easy digestion by humans, using a computer program. This approach shortens the information content of a text file while preserving the original contents. Text summarization has become an important and timely tool for assisting and interpreting text information in today's fast-growing information age. This task is essentially a data reduction process. The goal of automatic text summarization is condensing the source text into a shorter version preserving its information content and overall meaning. Automatic document summarization is an important research area in natural language processing (NLP). As The problem of information overload has grown, and as the quantity of data has increased, so has interest in automatic summarization. It is very difficult for human beings to manually summarize large documents of text. Text Summarization methods can be classified into abstractive and extractive summarization.

Abstractive summarization aims at paraphrasing the source document, similar to manual summarization. An extractive summarization method consists of selecting important sentences, paragraphs etc. from the original document and concatenating them into smaller form. The importance of sentences is decided based on statistical and linguistic features of sentences. Extractive methods work by selecting a subset of existing words, phrases, or sentences in the original text to form the summary. The extractive summarization systems are typically based on techniques for sentence extraction and aim to cover the set of sentences that are most important for the overall understanding of a given document.

The summarization has been studied by the Natural Language Processing community for nearly the last half period. The simple definition provides three important aspects that characterize research on automatic summarization: Summaries may be produced from a single document or multiple documents; Summaries should preserve important information; Summaries should be short. Text summarization is a process of producing a reduced version of original text that highlights the important contents of the text. It is an information retrieval task.

The important functions of the summarizer are:

- Reducing a single document to a user-defined fraction of its original size while maintaining coherence.
- Choosing the most relevant and important sentences from the text.
- Improving the abstraction and length of the summary by using a thesaurus to replace semantically related units.

There are many methods to proceed with automatic text summarization. In this model an extractive technique to obtain the summary from the given text. This summary is then improved further by replacing a few parts of it using an abstractive technique. The extraction of sentences from

the document is done keeping consistency in mind and therefore the summary maintains the core of the original document. The sentences are then ranked using a text-ranking algorithm and the final cluster or summary is formed.

The major application areas, where automatic text summarization is used, are:-

- Search engines – to present compressed descriptions of the search results to the user so that the user can read and understand the content of the retrieved documents.
- Document summarization – to store only the summarized version rather than the whole document.
- Text to Speech application – a text to speech application can use summaries rather than the whole document, since written text can be too long to listen to and time taking while the main information can be transmitted to the listener from the summary of the text.
- Small Screen Devices – summaries best fit small screen applications like in smart phones and iPods rather than having the whole document which might be too long in the small screen devices.

In effect, we aim to extractive summarize a single English document, not more than 300 sentences long, to a fraction of its original size, while maintaining cohesion, and then use a lexical database to abstract the generated summary. The software uses the external tool WordNet to abstract the generated summary. WordNet is a lexical database that groups words by semantic relations..

2. Background

2.1 Definition

A summary can be defined as a text that is produced from one or more texts, that contain a significant portion of the information in the original text, and that is no longer than half of the original text. Text summarization is the process of distilling the most important information from a source to produce a concise version for a particular user and task.

When this is done by means of a computer, i.e. automatically, we call this Automatic Text Summarization. Despite the fact that text summarization has traditionally been focused on text input, the input to the summarization process can also be multimedia information, such as images, video or audio, as well as on-line information or hypertexts. Furthermore, we can talk about summarizing only one document or multiple ones. In that case, this process is known as Multi-document Summarization (MDS) and the source documents in this case can be in a single-language or in different languages.

The output of a summary system may be an extract (i.e. when a selection of "significant" sentences of a document is performed) or abstract, when the summary can serve as a substitute to the original document. We can also distinguish between generic summaries and user-focused summaries. The first type of summaries can serve as surrogate of the original text as they may try to represent all relevant features of a source text. They are text-driven and follow a bottom-up approach using IR techniques. The user-focused summaries rely on a specification of a user information need, such a topic or query. They follow a top-down approach using IE techniques.

Traditionally, summarization has been decomposed into three main stages which is:

- **Interpretation** of the source text to obtain a text representation.
- **Transformation** of the text representation into a summary representation.
- **Generation** of the summary text from the summary representation Effective summarizing requires an explicit and detailed analysis of context factors. Three classes of context factors: input, purpose and output factors

2.2 Process of Automatic Text Summarization

Extractive summaries do not focus on the understanding of text. It extracts the most important part based on statistical and linguistic features such as cue words, location and word frequency.

The processing phase of summarization is a structural framework of the text. It consists of:

- Sentence boundary identification: - identification of boundary is identified by the dot at the termination of a sentence.
- Stop word elimination: - stop words and unnecessary information is discarded.
- Stemming: - for every word a stem is build which gives meaning.

In other words, first clean the text file by removing full stop, common words (conjunction, verb, adverb, preposition etc.). Then calculate the frequency of each word and select top words which have maximum frequency. This technique retrieves important sentence emphasize on high information richness in the sentence as well as high Information retrieval. These related maximum sentence generated scores are clustered to generate the summary of the document. Thus we use k-mean clustering to these maximum sentences of the document and find the relation to extract clusters with most relevant sets in the

document, these helps to find the summary of the document.

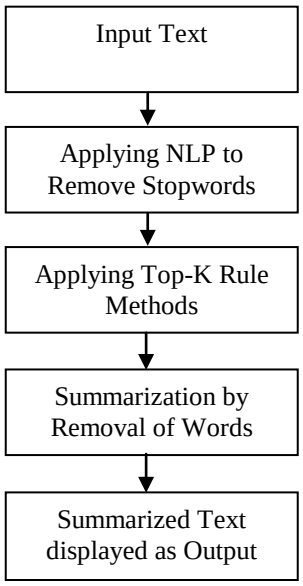

Fig. 1 Process on text summarization using top-k rules

3. Preliminaries

3.1 Existing System

Previous work on text summarization is limited to natural language processing based approach. This approach is good for sentence level classification but might not give accurate results when applied to an entire paragraph. Thus the efficiency of the existing system might be less as compare to the system which is based on rule mining.

3.2 Proposed System

Our system works on summarization technique based on rule mining.Automatic summarization is the process of condensing textual content into a concise form for easy digestion by humans, using a computer program. It is used to condense the large amounts of textual data. The purpose of the project is to provide a faster way of analysing sentences without losing the effect of grammatical structures, or the semantic and syntactic information that have been applied to or extracted from the program. In this approach, we integrate natural language processing with rule mining so that, the advantages of both the techniques can be combined to create an automatic text summarizer. By using top k rules based approach to find out the support and confidence of text parts which appeared more frequently in the input dataset. And also by using Jaccard distance to find similarity between two sets. This will allow us to find the best possible summary of documents which will be grammatically and content wise more accurate. Thus, the overall efficiency of the system is increased.

4. Methodologies

Our work has been composed in following steps:

1. Collection of input dataset for text mining.
2. Application of natural language processing the data set.
3. Finding similarity using Jaccard distance between two sets.
4. Application of Top-K Rule for finding out the best possible combination of words appearing in the input.
5. Integration of NLP with rule mining to find out summary of the document.
6. Result evaluation and comparison.

4.1 Natural Processing Task

NLP performs two ways in this paper:

- **Part-Of-Speech Tagging** (POS) aims at labelling each word with a unique tag that indicates its syntactic role, e.g. plural noun, adverb, etc.
- **Chunking**, also called shallow parsing, aims at labelling segments of a sentence with syntactic constituents such as noun or verb phrase (NP or VP). Each word is assigned only one unique tag, often encoded as a begin-chunk or inside-chunk tag

4.2 Jaccard Distance Method

Jaccard Distance Algorithm is implemented is as follows:

Jaccard Index is also known as the Jaccard similarity coefficient. Jaccard index is used for comparing the similarity and diversity of sample sets.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

Jaccard distance measures how dissimilar two sample sets are. As the formula show, Jaccard distance is the complementary to the Jaccard coefficient.

$$d_J(A,B) = 1 - J(A,B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|}$$

Jaccard index and Jaccard distance measure the overlap of binary attributes in A and B.

M_{11} represents the total number of attributes where A and B both have a value of 1.

M_{01} represents the total number of attributes where the attribute of A is 0 and the attribute of B is 1.

M_{10} represents the total number of attributes where the attribute of A is 1 and the attribute of B is 0.

M_{00} represents the total number of attributes where A and B both have a value of 0.

The Jaccard similarity coefficient, *J*, is given as

$$J = \frac{M_{11}}{M_{01} + M_{10} + M_{11}}$$

The Jaccard distance, *J'*, is given as

$$J' = \frac{M_{01} + M_{10}}{M_{01} + M_{10} + M_{11}}$$

4.3 Top-K Rule Method

The Top-K Rules algorithm takes as input a transaction database, a number *k* of rules that the user wants to discover, and the *minconf* threshold.

The algorithm main idea is the following. Top-K Rules first sets an internal *minsup* variable to 0. Then, the algorithm starts searching for rules. As soon as a rule is found, it is added to a list of rules L ordered by the support. The list is used to maintain the top-k rules found until now. Once *k* valid rules are found, the internal *minsup* variable is raised to the support of the rule with the lowest support in L. Raising the *minsup* value is used to prune the search space when searching for more rules. Thereafter, each time a valid rule is found, the rule is inserted in L, the rules in L not respecting *minsup* anymore are removed from L, and *minsup* is raised to the value of the least interesting rule in L. The algorithm continues searching for more rules until no rule are found, which means that it has found the top-k rules.

To search for rules, Top-K Rules does not rely on the classical two steps approach to generate rules because it would not be efficient as a top-k algorithm. The strategy used by Top-K Rules instead consists of generating rules containing a single item in the antecedent and a single item in the consequent. Then, each rule is recursively grown by adding items to the antecedent or consequent. To select the items that are added to a rule to grow it, Top-K Rules scans the transactions containing the rule to find single items that could expand its left or right part. We name the two processes for expanding rules in Top-K Rules *left expansion* and *right expansion*. These processes are applied recursively to explore the search space of association rules.

Another idea incorporated in TopKRules is to try to generate the most promising rules first. This is because if rules with high support are found earlier, TopKRules can raise its internal *minsup* variable faster to prune the search space. To perform this, TopKRules uses an internal variable R to store all the rules that can be expanded to have a chance of finding more valid rules. TopKRules uses this set to determine the rules that are the most likely to produce valid rules with a high support to raise *minsup* more quickly and prune a larger part of the search space.

Top-K Rule Algorithms as follows:

The main procedure of TopKRules is shown in Figure 2. The algorithm first scans the database once to calculate *tids*({*c*}) for each single item *c* in the database (*line 1*). Then, the algorithm generates all valid rules of size 1*1 by considering each pair of items *i, j*, where *i* and *j* each have at least *minsup*×|T| tids (if this condition is not met, clearly, no rule having at least the minimum support can be created with *i, j*) (line *2*). The supports of the rules {*i*}→{*j*} and {*j*}→{*i*} are simply obtained by dividing |*tids*(*i*→ *j*)| by |T| and |*tids*(*j*→ i)| by |T| (*line 3 and 4*). The confidence of the rules {*i*}→{*j*} and {*j*}→{*i*} is obtained by dividing |*tids*(*i*→ *j*)| by |*tids*(*i*)| and |*tids*(*j*→ *i*)| by | *tids*(*j*)| (line *5 and 6*). Then, for each rule {*i*}→{*j*} or {*j*}→{*i*} that is valid, the procedure SAVE is called with the rule and L as parameters so that the rule is recorded in the set L of the current top-k rules found (line *7 to 9*). Also, each rule {*i*}→{*j*} or {*j*}→{*i*} that is frequent is added to the set R, to be later considered for expansion and a special flag named *expandLR* is set to true for each such rule (line *10 to 12*).

TOPKRULES(T, *k, minconf*) R := Ø. L := Ø. *minsup* := 0.
1. Scan the database T once to record the tidset of each item.
2. FOR each pairs of items *i, j* such that |*tids(i)*| ×|T| ≥ *minsup* and |*tids*(*j*)| ×|T| ≥ *minsup*:
3. *sup*({*i*}→{*j*}) := |*tids(i)* ∩ *tids*(*j*)| / |T |.
4. *sup*({*j*}→{*i*}) := |*tids(i)* ∩ *tids*(*j*)| / |T|.
5. *conf*({*i*}→{*j*}) := |*tids(i)* ∩ *tids*(*j*)| / |*tids*(*i*)|.
6. *conf*({*j*}→{*i*}) := |*tids(i)* ∩ *tids*(*j*)| / |*tids*(*j*)|.
7. IF *sup*({*i*}→{*j*}) ≥ minsup THEN
8. IF *conf*({*i*}→{*j*}) ≥ *minconf* THEN SAVE({*i*}→{*j*}, L, *k, minsup*).
9. IF *conf*({*j*}→{*i*}) ≥ *minconf* THEN SAVE({*j*}→{*i*}, L, *k, minsup*).
10. Set flag expandLR of {*i*}→{*j*}to true.
11. Set flag expandLR of {*j*}→{*i*}to true.
12. R: = R∪{{*i*}→{*j*}, {*j*}→{*i*}}.
13. END IF
14. END FOR
15. WHILE ∃*r* ∈ R AND *sup*(*r*) ≥ *minsup* DO
16. Select the rule *rule* having the highest support in R
17. IF *rule*.expandLR = true THEN
18. EXPAND-L(*rule*, L, R, *k, minsup, minconf*).
19. EXPAND-R(*rule*, L, R, *k, minsup, minconf*).
20. ELSE EXPAND-R(*rule*, L, R, *k, minsup, minconf*).
21. REMOVE *rule* from R.
22. REMOVE from R all rules *r* ∈ R | *sup*(*r*) <*minsup*.
23. END WHILE

5. Evaluation Result

Original Text

BankAmerica Corp is not under pressure to act quickly on its proposed equity offering and would do well to delay it because of the stock's recent poor performance, banking analysts said.

Some analysts said they have recommended BankAmerica delay it's up to one-billion-dlr equity offering, which has yet to be approved by the Securities and Exchange Commission.

BankAmerica stock fell this week, along with other banking issues, on the news that Brazil has suspended interest payments on a large portion of its foreign debt.

The stock traded around 12, down 1/8, this afternoon, after falling to 11-1/2 earlier this week on the news.

Banking analysts said that with the immediate threat of the First Interstate Bancorp & lt takeover bid gone, BankAmerica is under no pressure to sell the securities into a market that will be nervous on bank stocks in the near term.

BankAmerica filed the offer on January 26. It was seen as One of the major factors leading the First Interstate withdrawing its takeover bid on February 9.

A BankAmerica spokesman said SEC approval is taking longer than expected and market conditions must now be re-evaluated. "The circumstances at the time will determine what we do, " said Arthur Miller, BankAmerica's Vice President for Financial Communications, when asked if BankAmerica would proceed with the offer immediately after it receives SEC approval.

"I'd put it off as long as they conceivably could," said Lawrence Cohn, analyst with Merrill Lynch, Pierce, Fenner and Smith.
Cohn said the longer BankAmerica waits, the longer they have to show the market an improved financial outlook.

Although BankAmerica has yet to specify the types of equities it would offer, most analysts believed a convertible preferred stock would encompass at least part of it.

Such an offering at a depressed stock price would mean a lower conversion price and more dilution to BankAmerica stock holders, noted Daniel Williams, analyst with Sutro Group.

Summarized Text

The stock traded around 12, down 1/8, this afternoon, after falling to 11-1/2 earlier this week on the news.

BankAmerica filed the offer on January 26.

It was seen as one of the major factors leading the First Interstate withdrawing its takeover bid on February 9.

A BankAmerica spokesman said SEC approval is taking longer than expected and market conditions must now be re-evaluated.

"The circumstances at the time will determine what we do," said Arthur Miller, BankAmerica's Vice President for Financial Communications, when asked if BankAmerica would proceed with the offer immediately after it receives SEC approval.

"I'd put it off as long as they conceivably could," said Lawrence Cohn, analyst with Merrill Lynch, Pierce, Fenner and Smith.

Cohn said the longer BankAmerica waits, the longer they have to show the market an improved financial outlook.

Although BankAmerica has yet to specify the types of equities it would offer, most analysts believed a convertible preferred stock would encompass at least part of it.

Such an offering at a depressed stock price would mean a lower conversion price and more dilution to BankAmerica stock holders, noted Daniel Williams, analyst with Sutro Group.

Output

Time needed: 343 ms

Initial Lines: 19

Final Lines: 12

Compression Ratio: 36.842106%

5.1 Compare Between Top-K Rule and Parsing Method

#	Input Lines	Time Taken (ms)		Output Lines	
		Parsing Method	Top-K Rule	Parsing Method	Top-K Rule
1	100	300	150	62	45
2	60	225	50	40	20

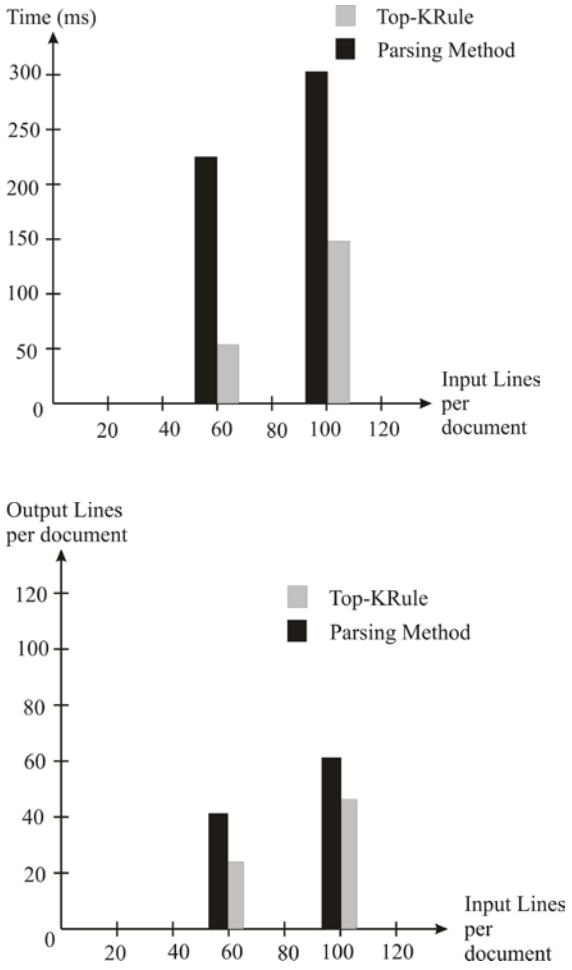

Fig. 1 Comparison between Top-K Rule and Parsing Method.

6. Conclusion

Our algorithm has proved to perform well for most summarization purposes. The current extractive summary is advantageous for certain formats of documents. Depending on the choice of parameters, association rule mining algorithms can generate an extremely large number of rules which lead algorithms to suffer from long execution time and huge memory consumption, or may generate few rules, and thus omit valuable information. To address this issue, we proposed Top-k Rules, an algorithm to discover the top-k rules having the highest support, where k is set by the user. To generate rules, Top-K Rules relies on a novel approach called rule expansions and also includes several optimizations that improve its performance. Experimental results show that Top-K Rules has excellent performance and scalability, and that it is an advantageous alternative to classical association rule mining algorithms when the user wants to control the number of association rules generated. The advantage of this method is that it operates completely algorithmically, and does not require sophisticated techniques. However, often the replacement is not sufficiently appropriate or ideal.

References

[1] Mihalcea, Rada, and Paul Tarau. “TextRank: Bringing order into texts.” Proceedings of EMNLP.Vol. 4.No. 4. 2004.

[2] Ravi Som Sinha and Rada Flavia Mihalcea, “Using centrality algorithms on directed graphs for synonym expansion.” FLAIRS Conference, AAAI Press, 2011.

[3] Blondel, Vincent D., and Pierre P. Senellart. “Automatic extraction of synonyms in a dictionary.” vertex 1 (2011): x1.

[4] Sankar, K., and L. Sobha. “An approach to text summarization.” Proceedings of the Third International Workshop on Cross Lingual Information Access: Addressing the Information Need of Multilingual Societies. Association for Computational Linguistics, 2009

[5] George A. Miller (1995). “WordNet: A Lexical Database for English.” Communications of the ACM Vol. 38, No. 11: 39-41. Christiane Fellbaum (1998, ed.) “WordNet: An Electronic Lexical Database.” Cambridge, MA: MIT Press.

[6] Lin, C. Y. (2004, July). “Rouge: A package for automatic evaluation of summaries.” In Text Summarization Branches Out: Proceedings of the ACL-04 Workshop (pp. 74-81)

[7] Bird, Steven, Edward Loper and Ewan Klein (2009), “Natural Language Processing with Python.” O’Reilly Media Inc.

[8] Sample Text Source: Grolier Electronic Publishing, Inc., 1995.

[9] H. Takamura and M. Okumura, “Text summarization model based on the budgeted median problem,” in Proc. 18th ACM Conf. Inf. Knowl.Manage., 2009, pp. 1589–1592, ACM.

[10] U. Hahn and U. Reimer, “Knowledge-based text summarization: Salience and generalization operators for knowledge base abstraction,” Adv. Automatic Text Summarization, pp. 215–232, 1999.

[11] A. Molina, “A study on sentence compression for the automatic summarization,” Ph.D. dissertation, Univ. d'Avignonet des Pays de Vaucluse (UAPV), Avignon, France, 2013.

[12] TAC, Tac 2011 guided summarization task guidelines. [Online].Available: http://www.nist.gov/tac/2011/Summarization/Guided-Summ.2011.guidelines.html 2011

[13] D. Radev, T. Allison, S. Blair-Goldensohn, J. Blitzer, A. Celebi, S.Dimitrov, E. Drabek, A. Hakim, W. Lam, and D. Liu et al., “Mead-a platform for multidocument multilingual text summarization,” in Proc.4th Int. Conf. Lang. Resources Eval.(LREC’04), 2004.

[14] D. Gillick, K. Riedhammer, B. Favre, and D. Hakkani-Tur, “A global optimization framework for meeting summarization,” in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP’09), 2009, pp.4769–4772.

[15] Karen Sparck Jones, "Automatic summarizing: factors anddirections, in: Advances in Automatic Text Summarization",MIT Press, pp. 1–12, 1999.

[16] Li Chengcheng, "Automatic text summarization based on rhetorical structure theory," Computer Application and System Modeling (ICCASM), 2010 International Conference. [Accessed 22-24 Oct], 2010.

[17] M. Rajman and R. Besancon, “Text mining: natural language techniques and text mining applications”, in *Proc. 7th working conf. on database semantics (DS-7), Chapan &Hall IFIP Proc. Series.* Leysin, Switzerland Oct. 1997, 7-10.

[18] Sparck Jones, K.: Discourse Modelling for Automatic Summarising. Technical Report No.290. University of Cambridge (1993).

[19] Oi Mean Foong, Alan Oxley and Suziah Sulaiman, “Challenges and Trends of Automatic Text Summarization”, IJITT, Vol. 1, Issue 1, ISSN: 0976–5972, 2010.

[20] Miller, S., Fox, H., Ramshaw, L., & Weischedel, R. (2000). A novel use of statistical parsing to extract information from text. 6th Applied Natural Language Processing Conference.