Text Summarization with Semantics Information

Namita Mittal
Department of Comp. Engg.
Malaviya National Institute of Technology, Jaipur
nmittal@mnit.ac.in

Basant Agarwal
Department of Comp. Engg.
Malaviya National Institute of Technology, Jaipur
thebasant@gmail.com

Nikita Vijay
Department of Comp. Engg.
Malaviya National Institute of Technology, Jaipur
niki.vijay26@gmail.com

Adarsh Gupta
Department of Comp. Engg.
Malaviya National Institute of Technology, Jaipur
guptaadarsh91@gmail.com

Abstract

Due to the wide use of Internet and the diversity of information, there is a large amount of information which is available to users. So many techniques have been developed for the access of large amount data quickly and accurately. Text summarization helps in reducing the size of a text while preserving its information content. One of the main drawbacks of Automatic Summarization is the vague semantic classification of the document, which results in the poor quality of the consequent summaries. In this paper, we propose an automatic summarization approach utilizing both Semantics and Text-Rank algorithm. Semantic graph-based approach is used for extractive summarization in order to solve the problem. The summarizer uses WordNet to produce a semantic graph that represents the document in such a way that edges between sentences are based on semantic similarity between sentences and the sentences are ranked by applying PageRank to the resulting graph. A summary is formed by selecting the top ranked sentences, using a threshold based on required size of the summary. ROUGE toolkit is used for evaluation of summaries DUC 2002 datasets and generates results.

Keywords: Text Summarization, Extractive Summarization, Text rank, Semantic Analysis

1. Introduction

Due to the continuous growth of data over World Wide Web, large amount of information is available to users. So many techniques have been developed for the access of large amount data quickly and accurately. Text summarization helps in reducing the size of a text while preserving its information content. Text summarization creates the compressed version of the input text without loss of information. The existing automatic summarization methods can be divided into two categories: extractive summarization and abstractive summarization. Extractive summarization extracts the important sentences from the original document to construct summary. Abstractive summarization based summaries are created by understanding the meaning of the document and then on that basis summary sentences are formed to construct the summary. Graph-based methods have attracted the attention of the NLP community which is applied to tasks such as word sense disambiguation or question answering (Plaza et al. 2011). These methods have typically tried to find important sentences in the text according to their similarity to other sentences. However, few approaches have tried to leverage the semantics of the text. This paper investigates the effect of incorporating semantic information for summarization.

Organization of paper is as follows. Section 2 covers the Related Work, Section 3 presents the Proposed Approach, Section 4 discusses experiments and results, and Section 5 concludes and presents the Future Work.

2. Related Work

Summarization models may be classified into extractive summary and abstractive summary (Chuang et al. 2000). Extractive summarization model creates extracts by selecting important sentences and another creates abstract by understanding the meaning of the whole text. In this paper, extractive method is focused, that is, those which select sentences from the original document to produce the summary. LexRank (Erkan and Radev, 2004), is a centroid-based method for multi-document summarization that computes the sentence importance based on the concept of eigenvector centrality. It assumes a fully connected, undirected graph with sentences as nodes using tf-idf score and similarities between them as edges using cosine similarity. A very similar system is TextRank (Mihalcea et al. 2004b), is also been proposed for single-document summarization. The following proposed algorithm is different from each of these and computes the semantic distance between the sentences in the document and then applies Text Rank algorithm. Most of the existing summarization does not use semantic content of the sentence and relative importance of the content to the semantic of the text. Proposed approach is based on identifying the semantic relations among sentences.

3. Proposed Approach

Main aim is to build automatic extractive summarizer by combining Google's page rank (Brin and Page, 1998) algorithm and Weighted Graph (WG) based representation of document by utilizing the semantic relations between the sentences. WG representation is a powerful and effective tool in which the rank carries the significance of a vertex (sentence) in the graph (document) by accounting all the global information from entire graph. Rank of the vertex at the end of each iteration is updated and the connection between sentences (edges) is derived based on similarity between sentences. The similarity measure is calculated on many parameters like content overlap, semantic measures. With long sentences weights can be normalized with respective sentence lengths (Brin and Page, 1998).

The proposed approach consists of 4 main steps:

i. Preprocessing of the text document,
ii. Sentence Processing,
iii. Sentences Semantic Similarity,
iv. Sentence Rank calculation and Sentence Extraction.

Each step is discussed in detail in the following subsections. Testing is done using documents collection of DUC 2002 and evaluated using ROUGE measures.

3.1 Preprocessing of the Text Document

Stanford Core-NLP library is used for preprocessing of documents for Tokenization, Part of Speech Tagger, Lemmatizer and Sentence Splitter annotator. In order to construct the concept graph that represents the document, the following preprocessing steps are undertaken:

1 To split the document into sentences using the **Sentence Splitter Annotator**.

2 **Stop-word elimination:** In this step, the features of alphabet tokens are identified namely as Determiner, Preposition, Noun, Verb, Adjective etc. Common words with no semantics and which do not give relevant information to the task (e.g., “the”, “a”) are eliminated.

3 **Case folding**: All the characters are converted into lower case.

4 **Lemmatization:** It is the algorithmic process of determining the lemma for a given word. It converts the word into its root format keeping context information into consideration.

Next, each sentence is transformed into appropriate concepts in WordNet, using different measures of semantic similarity and relatedness to perform WSD and then assigns a sense (as found in WordNet) to each word in a text. In this work, the Adapted Lesk WSD method is used, which computes semantic relatedness of word senses using gloss overlaps (Banerjee et al. 2002).

3.2 Sentence Processing

After preprocessing of the document, sentences are received and treated as node of graph. These sentences will be processed individually for further steps of summarization. Noun (n), Verb(v), Adj(a), Adverb(r) are stored for all the individual sentences.

3.3 Sentences Semantic Similarity

To this end, similarities between every pair of leaf concepts in the graph are determined using the WordNet. WordNet package implements a variety of semantic similarity and relatedness measures. In the experiments, Adapted Lesk measure is used. To expand the document graph with these additional relations, a new edge is added between two leaf nodes if the similarity between the underlying concepts exceeds a similarity threshold. Finally, each edge is assigned a weight in [0, 1]. This edge weight is computed as the ratio between the relative positions in their corresponding hierarchies of the concepts linked by the edge. And thus sentence to sentence similarity scores are got by using this Adapted Lesk algorithm.

3.4 Sentence Rank Calculation and Sentence Extraction

In this step, tokenization of the document is performed using above mentioned preprocessing methods by neglecting all unnecessary weightless noise and extracting individual tokens to be forwarded for lexicon analysis.

3.4.1 Page Rank

PageRank (Brin and Page, 1998) is one of the most popular ranking algorithms, PageRank integrates the impact of both incoming and outgoing links into one single model, and therefore it produces only one set of scores (Mihalcea et al. 2004a).

3.4.2 Text Rank for Sentence Extraction

To apply TextRank, firstly graph associated with the text is build, where the graph vertices are representative for the units to be ranked. The main goal is to rank entire sentences; therefore the vertex is added to the graph for each sentence of the text. Next, connection between two sentences is determined by similarity relations between them, and similarity is measured by content overlap. A link is drawn between two sentence nodes if they share mostly common content. The measure of content overlap is determined by semantic similarity algorithm discussed in previous steps. To avoid long sentences, a normalization factor is used, and divides the content overlap by it. The resulting graph which is produced of sentences as vertex and edges representing the similarities with a weight associated with each edge. The text is therefore represented as a weighted graph, and consequently the weighted graph-based ranking formula is used as discussed in Page Rank algorithm section. After the ranking algorithm run on the graph, top ranked sentences are selected for the summary on the basis of their scores. Figure1 showed a weighted graph with weights attached to the edges, and the final TextRank score computed for each sentence. The sentences with the highest rank are selected to include in the summary.

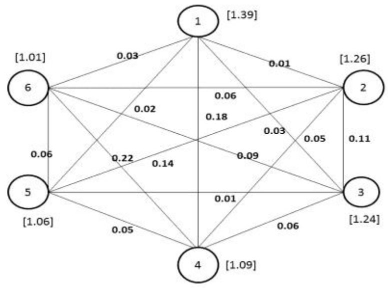

Figure 1 :Weighted undirected graph for a document

3.4.3 Redundant Sentence Removal

To eliminate the redundant sentences from the retrieved ranked sentences, the general idea is to penalize the sentences which have high similarity with the one already in the summary. Assume that *S* is the set of sentences in the final summary and *C* is the set of candidate sentences. The algorithm process is as follows:

1. Initialize the two sets *S* = ∅ and C={si| i=1,2,3..,n} having all the extracted sentences.
2. Sort the sentences in *C* on the basis of the scores of the sentences.
3. Select the top ranked sentence in *C*. Move si from *C* to *S* and update the score of each remaining sentence *j* in *C* , if the similarity between the 2 sentences is less than threshold, then include both of them else include the one with higher score.
4. Repeat step (2) and step (3) until the number of selected sentences reach the summary length.

Thus, the top candidate sentences of key sentences are selected from the text based upon their relevance to the document, however, two sentences are similar to each other in terms of semantic content should not both selected.

4 Experiments and Results

Text Rank sentence extraction algorithm is evaluated in the context of a single-document summarization task, using the Document Understanding Evaluations 2002 datasets (DUC, 2002). For evaluation, the ROUGE 1.5.5 evaluation toolkit method is used, which uses N-gram statistics. Manually produced reference summaries are provided, and used in the evaluation process. The documents were given as an input to the summarizer system to produce their respective *System* summaries. *System* summaries are those that are produced by the system. On the other hand, *Gold* or *Model* summaries are the reference summaries. Recall, Precision and F-Measure values for 3 documents are given in Table 1that are tested.

Doc-id	Recall	Precision	F-Measure
AP880911-0016	0.379	0.359	0.369
AP880916-0060	0.705	0.411	0.519
WSJ880912-0064	0.586	0.469	0.521

Table 1-Precision, Recall and F-Measure for the documents with given Doc-id

Evaluation Results show that the Text Rank algorithm used gives the precision scores as shown in Table 1. Table 2 proves that our results are in coherence with them and the summaries are accurate.

Text Summarization Algorithms	**ROUGE-L (Precision)**
Sentence Rank	0.462
Text Rank- Page Rank	***0.500***
Lex Rank	0.469
MEAD	0.472
Sum Graph	0.484

Table 2: Comparison of precision scores on DUC 2002 dataset and ROUGE-L precision scores

Thus, the Text Rank approach to sentence extraction succeeds to select the most important sentences on the basis of the information provided by the document itself. TextRank is fully unsupervised, and relies only on the given text to derive an extractive summary. Among all algorithms, Page Rank algorithms provide the best performance, at par with the best performing system from DUC 2002. Text Rank goes beyond the sentence connectivity in a text. Another important advantage of Text Rank is that it gives a ranking over all sentences in a text which means that it can be easily adapted to extract very short summaries, or longer summaries, consisting of more than 100 words (Mihalcea et al. 2004a).

5 Conclusion

The goal of this research is to study the interaction between a set of statistical and semantic features and their impact on the process of extractive text summarization with the final objective of selecting the most significant features. The obtained results have shown that semantic-based methods stop word removal, lemmatization, POS tagging and word sense disambiguation improves the resultant summary. Redundant information is also detected to produce more accurate results. We evaluated experiments on DUC 2002 datasets. For the evaluation of the results, ROUGE scores are used. Comparison of the results against other text summarization approaches is also done. The results show that the Text Rank (Page Rank) algorithm gives perfect results when tested on DUC 2002 datasets. The future work will continue for the development of the Anaphora resolution module on the level of data representation. This will allow us to carry out the redundancy detection on the concept level in the process of abstractive Text Summarization.

References

Plaza L., Del A. 2011. “Using Semantic Graphs and Word Sense Disambiguation Techniques to Improve Text Summarization”, pp 97-105

Brin S. and Page L.. 1998. “The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN Systems”, 30(1–7).

Mihalcea R. 2004a. Graph-based ranking algorithms for sentence extraction, applied to text summarization. In Proceedings of the 42nd Annual Meeting of the Association for Computational Lingusitics (ACL), pp 170-173

Mihalcea, R. and Tarau, T. 2004b. TextRank – bringing order into texts. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 404–411.

Banerjee S. and Pedersen T. 2002. “An adapted lesk algorithm for word sense disambiguation using wordnet.” In 3rd International Conference on Intelligent Text Processing and Computational Linguistics, CICLing 2002, pp 136-145.

Erkan G. and Radev D.R. (2004) "LexRank: Graph-based Lexical Centrality as Salience in Text Summarization", In Journal of Artificial Intelligence Research, Volume 22, pages 457-479.

Chuang, W., Yang, J. (2000). Extracting Sentence Segments for Text Summarization: A Machine Learning Approach. In 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 152–159.