Variations of TextRank for Automated Summarization

Abstract. This article describes our proposal for new variants of the TextRank algorithm for automated summarization of texts. We describe the generalities of the TextRank algorithm on its original version and the different variations of the algorithm that we created. Some of these variants achieve a significative improvement over the original algorithm using the same metrics and dataset as the original publication.

Keywords: TextRank variations, automated summarization, Information Retrieval ranking functions

1 Introduction

We can describe the process of automated summarization as the extraction of the most important sentences in a document. Using different levels of compression a summarized version of the document of arbitrary length can be obtained. The TextRank algorithm is one of the most used methods for this task. TextRank builds a graph of sentences for a document and then applies PageRank to obtain a score for each sentence. In this article we describe several different proposals for the construction of the TextRank graph and report the results obtained with them.

The first section of this article describes previous work in the area and the TextRank algorithm in general. Then we report the results obtained for the different variations of the algorithm. Finally we describe the different metrics used for the evaluation of the results obtained from the proposed changes and the datasets used for these tests.

2 Previous work

The field of automated summarization has attracted interest since the late 50's [7]. Traditional methods for text summarization analyze the frequency of words or sentences in the first paragraphs of the text to identify the most important lexical elements. Several statistical models have been developed based on training corpora to combine different heuristics using keywords, position and length of sentences, word frequency and titles [6]. On a different approach, some algorithms analyze the semantic structure of the different textual units in a document with the goal of separating those that take a significative role in the representation of the document [8].

Other methods are based in the representation of the text as a graph: the most important sentences are the most connected ones in the graph and are used for

building a final summary [1]. These algorithms use different information retrieval techniques to identify similar sentences and determine the most important ones [12]. The TextRank algorithm developed by Mihalcea and Tarau [9] and the LexRank algorithm by Erkan and Radev [4] are based in ranking the lexical units of the text (sentences or words).

3 TextRank

3.1 Description

TextRank is an unsupervised algorithm for the automated summarization of texts that can also be used to obtain the most important keywords in a document. It was introduced in 2004 by Rada Mihalcea and Paul Tarau in [9].

The algorithm applies a variation of PageRank [10] over a graph constructed specifically for the task of summarization. This method provides an understanding of the structure of the document identifying its principal concepts without the need of previous training. Since the algorithm is based on PageRank it uses the idea of ranking of the elements in the graph, the most important elements are the ones that better describe the text. This approach allows TextRank to build summaries without the need of a training corpus or labeling and allows the use of the algorithm with different languages as long as there is a way to build the graph of sentences for it.

3.2 Text as a Graph

For the task of automated summarization, TextRank models any document as a graph using sentences as nodes [2]. A function to compute the similarity of sentences is needed to build edges in between. This function is used to weight the graph edges, the higher the similarity between sentences the more important the edge between them will be in the graph. In the domain of a Random Walker, as used frequently in PageRank, we can say that we are more likely to go from one sentence to another if those sentences are very similar.

The similarity function can use several different ideas. It can be based on the semantic of the sentences, on their proximity in the text, common words, and many other different metrics. The goal of this article is to experiment with these functions and report the results obtained when used along with TextRank.

The function featured in the original algorithm can be formalized as:

Definition 1 *Given S_i, S_j two sentences represented by a set of n words that in S_i are represented as $S_i = w_1^i, w_2^i, ..., w_n^i$. The similarity function for S_i, S_j can be defined as:*

$$Sim(S_i, S_j) = \frac{|\{w_k | w_k \in S_i \& w_k \in S_j\}|}{log(|S_i|) + log(|S_j|)} \tag{1}$$

The result of this process is a dense graph representing the document. From this graph, PageRank is used to compute the importance of each vertex. The most significative sentences are selected and presented in the same order as they appear in the document as the summary.

4 Example

A sample document from the 2002 DUC (Document Understanding Conference) dataset [3] is shown in Figure 1. The TextRank algorithm produces the summary shown in Figure 2 using the graph shown in Figure 3.

Fig. 1. Sample document from the DUC 2002 corpus.

1. Growth Factor Protects Heart Following Attack, Study In Rats Shows
2. By PAUL RECER AP Science Writer WASHINGTON (AP)
3. A natural substance called transforming growth factor beta appears to be able to limit damage to cardiac cells following a heart attack, according to a study published in the journal Science.
4. In a study at the Jefferson Medical College in Philadelphia, a group of laboratory rats induced to have heart attacks suffered 50 percent less cell damage after injections of transforming growth factor beta than did rats that did not receive the TGF beta.
5. "TGF beta is a growth factor that opposes some of the bad guys following a heart attack," said Dr. Allan Lefer, a professor at Jefferson.
6. Lefer said his research team simulated heart attacks in 24 rats by partially blocking key arteries in their hearts.
7. In 12 of the rats, the researchers injected a placebo.
8. In the other 12, they injected transforming growth factor beta.
9. For those who received the TGF beta, said Lefer, "the damage from the attack was much less severe.
10. There was about 50 percent less injury with TGF beta than without it".
11. The extent of heart cell damage was determined by measuring the amount of creatine kinase in the heart tissue following an attack.
12. Hearts damaged when the blood supply is interrupted, as in a heart attack, tend to lose creatine kinase, said Lefer.
13. Thus, by measuring for the loss of this substance, researchers could determine the amount of heart damage.
14. Lefer said the TGF beta seems to block the action of other substances, such as tumor necrosis factor, that can cause blood vessels to narrow following a heart attack.
15. Narrowed blood vessels carry less oxygen-rich blood to cells and this causes additional injury following a heart attack.
16. TGF beta is normally present in heart cells, but the study published in Science said that it is missing from rat heart cells damaged by a simulated heart attack.
17. Though TGF beta is produced naturally in the body, Lefer said his research team used a substance produced artificially by Genentech, a California biotechnology firm.
18. Lefer said his team is now conducting additional studies with TGF beta and that any experimental treatment of human heart attack victims with the substance is at least a year away.
19. Jefferson Medical College, where the study was done, is part of Thomas Jefferson University in Philadelphia.
20. Science, which published the study, is the journal of the American Association for the Advancement of Science.

Fig. 2. Summary generated from the sample text.

A natural substance called transforming growth factor beta appears to be able to limit damage to cardiac cells following a heart attack, according to a study published in the journal Science. In a study at the Jefferson Medical College in Philadelphia, a group of laboratory rats induced to have heart attacks suffered 50 percent less cell damage after injections of transforming growth factor beta than did rats that did not receive the TGF beta. Lefer said the TGF beta seems to block the action of other substances, such as tumor necrosis factor, that can cause blood vessels to narrow following a heart attack. TGF beta is normally present in heart cells, but the study published in Science said that it is missing from rat heart cells damaged by a simulated heart attack.

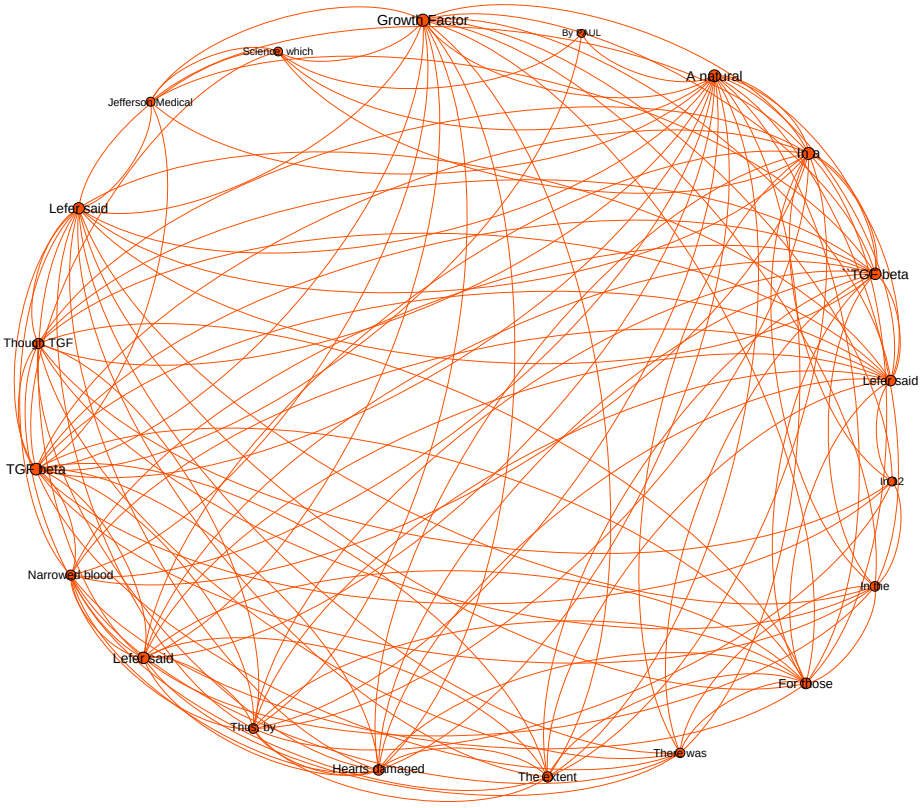

Fig. 3. Graph created from the sample text.

5 Experiments

5.1 Our Variations

This section will describe the different variations that we propose over the original TextRank algorithm. These ideas are based in changing the way in which

distances between sentences are computed to weight the edges of the graph used for PageRank. We found some of these variations to produce significative improvements over the original algorithm.

Longest Common Substring From two sentences we identify the longest common substring (LCS) and report the similarity to be its length.

Cosine Distance The cosine similarity is a metric widely used to compare texts represented as vectors. We used a classical TF-IDF model to represent the documents as vectors and computed the cosine between vectors as a measure of similarity. Since the vectors are defined to be positive, the cosine results in values in the range [0,1] where a value of 1 represents identical vectors and 0 represents orthogonal vectors.

BM25 BM25 / Okapi-BM25 is a ranking function widely used as the state of the art for Information Retrieval tasks. BM25 is a variation of the TF-IDF model using a probabilistic model.

Definition 2 *Given two sentences R, S, BM25 is defined as:*

$$BM25(R,S) = \sum_{i=1}^{n} IDF(s_i) \cdot \frac{f(s_i,R) \cdot (k_1+1)}{f(s_i,R) + k_1 \cdot (1 - b + b \cdot \frac{|R|}{avgDL})} \tag{2}$$

where k and b are parameters. We used $k = 1,2$ and $b = 0,75$. $avgDL$ is the average length of the sentences in our collection.

This function definition implies that if a word appears in more than half the documents of the collection, it will have a negative value. Since this can cause problems in the next stage of the algorithm, we used the following correction formula:

$$IDF(s_i) = \begin{cases} log(N - n(s_i) + 0.5) - log(n(s_i) + 0.5) & \text{if } n(s_i) > N/2 \\ \varepsilon \cdot avgIDF & \text{if } n(s_i) \leq N/2 \end{cases} \tag{3}$$

where ε takes a value between 0.5 and 0.30 and $avgIDF$ is the average IDF for all terms.

We also used BM25+, a variation of BM25 that changes the way long documents are penalized.

5.2 Evaluation

For testing the proposed variations, we used the database of the 2002 Document Understanding Conference (DUC) [3]. The corpus has 567 documents that are summarized to 20% of their size, and is the same corpus used in [9].

To evaluate results we used version 1.5.5 of the ROUGE package [5]. The configuration settings were the same as those in DUC, where ROUGE-1, ROUGE-2 and ROUGE-SU4 were used as metrics, using a confidence level of 95% and applying stemming. The final result is an average of these three scores.

To check the correct behaviour of our test suite we implemented the reference method used in [9], which extracts the first sentences of each document. We found the resulting scores of the original algorithm to be identical to those reported in [9]: a 2.3% improvement over the baseline.

5.3 Results

We tested LCS, Cosine Sim, BM25 and BM25+ as different ways to weight the edges for the TextRank graph. The best results were obtained using BM25 and BM25+. We achieved an improvement of 2.92% above the original TextRank result using BM25 and $\varepsilon = 0.25$. The following chart shows the results obtained for the different variations we proposed.

Table 1. Evaluation results for the proposed TextRank variations.

Method	ROUGE-1	ROUGE-2	ROUGE-SU4	Improvement
BM25 (Neg to epsilon)	0.4042	0.1831	0.2018	2.92%
BM25+ (Neg to epsilon)	0.404	0.1818	0.2008	2.60%
Cosine TF-IDF	0.4108	0.177	0.1984	2.54%
BM25+ (IDF = log(N/NI))	0.4022	0.1805	0.1997	2.05%
BM25 (IDF = log(N/NI))	0.4012	0.1808	0.1998	1.97%
Longest Common Substring	0.402	0.1783	0.1971	1.40%
BM25+ (Neg to zero)	0.3992	0.1803	0.1976	1.36%
BM25 (Neg to zero)	0.3991	0.1778	0.1966	0.89%
TextRank	**0.3983**	**0.1762**	**0.1948**	–
BM25	0.3916	0.1725	0.1906	-1.57%
BM25+	0.3903	0.1711	0.1894	-2.07%
DUC Baseline	0.39	0.1689	0.186	-2.84%

The result of Cosine Similarity was also satisfactory with a 2.54% improvement over the original method. The LCS variation also improved the original TextRank algorithm with 1.40% total improvement.

The performance in time was also improved. We could process the 567 documents from the DUC2002 database in 84% of the time needed in the original version.

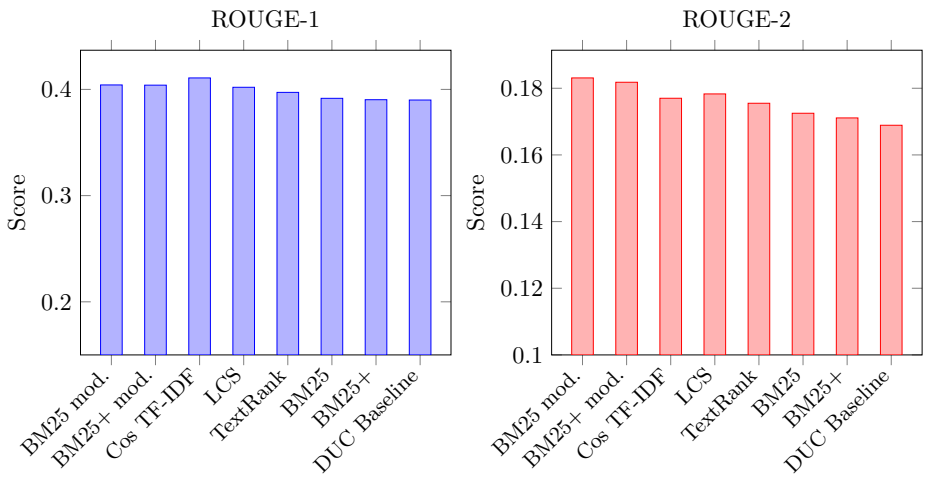

Fig. 4. ROUGE-1 and ROUGE-2 scores comparison.

6 Reference Implementations and Gensim Contribution

A reference implementation of our proposals was coded as a Python module. It can be obtained for testing and to reproduce results from an URL that will be provided in the non-anonymous version.

We also contributed the BM25-TextRank algorithm to the Gensim project [11].

7 Conclusions

This work presented three different variations to the TextRank algorithm for automatic summarization. The three variations presented improved significantly the results of the algorithm using the same metric and database used in the original paper. Given that TextRank performs 2.84% over the baseline, our improvement of 2.92% over the TextRank result is an important result.

The combination of TextRank with modern Information Retrieval ranking functions such as BM25 and BM25+ results in the creation of a robust method for automatic summarization that performs better than the standard techniques used previously.

Based on this results we suggest the use of BM25 along with TextRank for the task of unsupervised automatic summarization of texts. The results obtained and the examples analyzed show that this variation is better than the original TextRank algorithm without a performance penalty.

References

1. Barzilay, R., McKeown, K.: Sentence fusion for multidocument news summarization. Computational Linguistics 31(3), 297–328 (2005), http://dblp.uni-trier.de/db/journals/coling/coling31.html#BarzilayM05
2. Christopher D. Manning, Prabhakar Raghavan, H.S.: Introduction to Information Retrieval. Cambridge University Press (2008)
3. Document Understanding Conference: Duc 2002 guidelines (July 2002), http://www-nlpir.nist.gov/projects/duc/guidelines/2002.html
4. Erkan, G., Radev, D.R.: Lexrank: Graph-based lexical centrality as salience in text summarization. J. Artif. Intell. Res. (JAIR) 22, 457–479 (2004), http://dblp.uni-trier.de/db/journals/jair/jair22.html#ErkanR04
5. Lin, C.Y.: Rouge: a package for automatic evaluation of summaries. In: Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004), Barcelona, Spain (2004)
6. Lin, C.Y., Hovy, E.H.: Identifying topics by position. In: Proceedings of 5th Conference on Applied Natural Language Processing. Washington D.C. (March 1997)
7. Luhn, H.P.: The automatic creation of literature abstracts. IBM J. Res. Dev. 2(2), 159–165 (Apr 1958), http://dx.doi.org/10.1147/rd.22.0159
8. Marcu, D.: The theory and practice of discourse parsing and summarization. Computational Linguistics 28(1), 81–83 (2000), http://dblp.uni-trier.de/db/journals/coling/coling28.html#Hahn02
9. Mihalcea, R., Tarau, P.: Textrank: Bringing order into texts. In: Lin, D., Wu, D. (eds.) Proceedings of EMNLP 2004. pp. 404–411. Association for Computational Linguistics, Barcelona, Spain (July 2004)
10. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. In: Proceedings of the 7th International World Wide Web Conference. pp. 161–172. Brisbane, Australia (1998), citeseer.nj.nec.com/page98pagerank.html
11. Řehůřek, R., Sojka, P.: Software Framework for Topic Modelling with Large Corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. pp. 45–50. ELRA, Valletta, Malta (May 2010), http://is.muni.cz/publication/884893/en
12. Salton, G., Singhal, A., Mitra, M., Buckley, C.: Automatic text structuring and summarization. Information Processing and Management 33(2), 193 – 207 (1997)