Variations of the Similarity Function of TextRank for Automated Summarization

Federico Barrios[1], Federico López[1], Luis Argerich[1], Rosa Wachenchauzer[1,2]

[1] Facultad de Ingeniería, Universidad de Buenos Aires,
Ciudad Autónoma de Buenos Aires, Argentina.
[2] Universidad Nacional de Tres de Febrero, Caseros, Argentina.
{fbarrios,fjlopez}@fi.uba.ar

Abstract. This article presents new alternatives to the similarity function for the TextRank algorithm for automated summarization of texts. We describe the generalities of the algorithm and the different functions we propose. Some of these variants achieve a significative improvement using the same metrics and dataset as the original publication.

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

1 Introduction

In the field of natural language processing, an extractive summarization task can be described as the selection 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.

TextRank is an extractive summarization algorithm widely used as a result of its applicability: it is highly portable to other domains, genres, or languages since it does not require deep linguistic knowledge, nor domain or language specific annotated corpora. The algorithm builds a graph with the sentences of a document and then applies PageRank to obtain a score for each sentence.

In this article we present 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 an overview of the TextRank algorithm. Then we report the results obtained using the variations. 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 [12]. 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. The mainstream research in this field emphasizes extractive approaches to summarization using statistical methods [3]. Several statistical models have been developed based on training corpora to combine different heuristics using keywords, position and length of sentences, word frequency or titles [11].

Other methods are based in the representation of the text as a graph. The graph-based ranking approaches consider the intrinsic structure of the texts instead of treating texts as simple aggregations of terms. Thus it is able to capture and express richer information in determining important concepts [16].

The selected text fragments to use in the graph construction can be phrases [5], sentences [12], or paragraphs [15]. Currently, many successful systems adopt the sentences considering the tradeoff between content richness and grammar correctness. According to these approach the most important sentences are the most connected ones in the graph and are used for building a final summary [1]. To identify relations between sentences (edges for the graph) there are several measures: overlapping words, cosine distance, query-sensitive similarity or combinations of the previous, with supervised learning functions [16].

Finally, these algorithms use different information retrieval techniques to determine the most important sentences (vertices) and build the summary [20]. The TextRank algorithm developed by Mihalcea and Tarau [14] and the LexRank algorithm by Erkan and Radev [6] are based in ranking the lexical units of the text (sentences or words) using variation of PageRank [17]. Other graph-based ranking algorithms such as HITS [9] or Positional Function [8] may be also applied.

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 by Rada Mihalcea and Paul Tarau in [14].

The algorithm applies a variation of PageRank [17] over a graph constructed specifically for the task of summarization. This produces a 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.

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 [17], we can say that we are more likely to go from one sentence to another if they are very similar.

TextRank determines the relation of similarity between two sentences based on the content that both share. This overlap is determined simply as the number of common lexical tokens between them. To avoid promoting long sentences, the similarity function also divides this relation with the lenght of each sentence.

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 Experiments

4.1 Our Variations

This section will describe the different modifications 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. These similarity measures are orthogonal to the TextRank model, thus they can be easily integrated into the algorithm. 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 and report the similarity to be its length [7].

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 [21].

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 [19].

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. Other corrective strategies were also tested, setting $\varepsilon = 0$ and using simpler modifications of the classic IDF formula.

We also used BM25+, a variation of BM25 that changes the way long documents are penalized [13].

4.2 Evaluation

For testing the proposed variations, we used the database of the 2002 Document Understanding Conference (DUC) [4]. The corpus has 567 documents that are summarized to 20% of their size, and is the same corpus used in [14].

To evaluate results we used version 1.5.5 of the ROUGE package [10]. 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 [14], which extracts the first sentences of each document. We found the resulting scores of the original algorithm to be identical to those reported in [14]: a 2.3% improvement over the baseline.

4.3 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+ with the corrective formula shown in equation 3. 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 ($\varepsilon = 0.25$)	0.4042	0.1831	0.2018	2.92%
BM25+ ($\varepsilon = 0.25$)	0.404	0.1818	0.2008	2.60%
Cosine TF-IDF	0.4108	0.177	0.1984	2.54%
BM25+ (IDF = $\log(N/N_i)$)	0.4022	0.1805	0.1997	2.05%
BM25 (IDF = $\log(N/N_i)$)	0.4012	0.1808	0.1998	1.97%
Longest Common Substring	0.402	0.1783	0.1971	1.40%
BM25+ ($\varepsilon = 0$)	0.3992	0.1803	0.1976	1.36%
BM25 ($\varepsilon = 0$)	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%

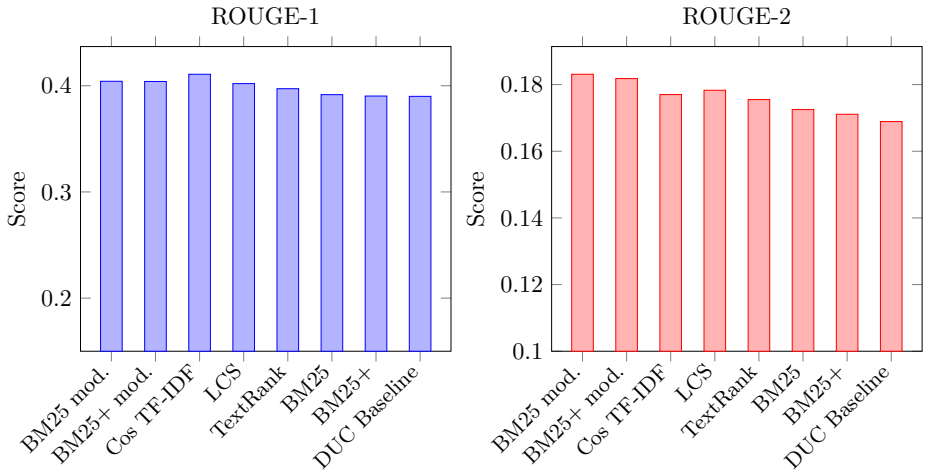

Fig. 1. ROUGE-1 and ROUGE-2 scores comparison.

The result of Cosine Similarity was also satisfactory with a 2.54% improvement over the original method. The LCS variation also performed better than 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.

5 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 [18].

6 Conclusions

This work presented three different variations to the TextRank algorithm for automatic summarization. The three alternatives improved significantly the results of the algorithm using the same test configuration as in the original publication. Given that TextRank performs 2.84% over the baseline, our improvement of 2.92% over the TextRank score is an important result.

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

Based on these 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. Das, D., Martins, A.F.T.: A survey on automatic text summarization. Tech. rep., Carnegie Mellon University, Language Technologies Institute (2007)
4. Document Understanding Conference: Duc 2002 guidelines (July 2002), http://www-nlpir.nist.gov/projects/duc/guidelines/2002.html
5. Ercan, G., Cicekli, I.: Using lexical chains for keyword extraction. Inf. Process. Manage. 43(6), 1705–1714 (Nov 2007), http://dx.doi.org/10.1016/j.ipm.2007.01.015
6. 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
7. Gusfield, D.: Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology. Cambridge University Press, New York, NY, USA (1997)
8. Herings, P.J.J., van der Laan, G., Talman, D.: Measuring the power of nodes in digraphs. Research Memorandum 007, Maastricht University, Maastricht Research School of Economics of Technology and Organization (METEOR) (2001), http://EconPapers.repec.org/RePEc:unm:umamet:2001007

9. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. J. ACM 46(5), 604–632 (Sep 1999), http://doi.acm.org/10.1145/324133.324140
10. 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)
11. 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)
12. 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
13. Lv, Y., Zhai, C.: Lower-bounding term frequency normalization. In: Proceedings of the 20th ACM Conference on Information and Knowledge Management, CIKM 2011, Glasgow, United Kingdom, October 24-28, 2011. pp. 7–16 (2011)
14. 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)
15. Mitrat, M., Singhal, A., Buckleytt, C.: Automatic text summarization by paragraph extraction. In: Intelligent Scalable Text Summarization. pp. 39–46 (1997), http://www.aclweb.org/anthology/W97-0707
16. Ouyang, Y., Li, W., Wei, F., Lu, Q.: Learning similarity functions in graph-based document summarization. In: Li, W., Aliod, D.M. (eds.) ICCPOL. Lecture Notes in Computer Science, vol. 5459, pp. 189–200. Springer (2009), http://dblp.uni-trier.de/db/conf/iccpol/iccpol2009.html#OuyangLWL09
17. 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
18. 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
19. Robertson, S., Walker, S., Jones, S., Hancock-Beaulieu, M., Gatford, M.: Okapi at TREC-3. In: Proceedings of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2-4, 1994. pp. 109–126 (1994), http://trec.nist.gov/pubs/trec3/papers/city.ps.gz
20. Salton, G., Singhal, A., Mitra, M., Buckley, C.: Automatic text structuring and summarization. Information Processing and Management 33(2), 193 – 207 (1997)
21. Singhal, A.: Modern information retrieval: A brief overview. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering 24(4), 35–43 (2001), http://singhal.info/ieee2001.pdf