ON THE OPEN TEXT SUMMARIZER

Filip Andonov, Velina Slavova, Georgi Petrov

Abstract*: This paper presents proposed improvements to the Open Text Summariser (OTS) based on a heuristic approach. It describes the ten steps of the implemented algorithm and outlines further ideas for its development. The authors discuss valuable feedback gained from four experts evaluating the summaries generated by the OTS And the improved OTS and propose paths for future improvements. Then the results of the experts' evaluation of the two summarizers are presented and analyzed. Problems affecting the precision of both summarizing algorithms are discussed.*

Keywords: *Summary, extraction-based method, heuristics*

ACM Classification Keywords*: 1.2.7. Natural language processing, 1G.3 Correlation and regression analysis,*

Introduction

The Open Text Summarizer (OTS) is an implementation of a grammar-agnostic method for creating a summary of a text. It is a simple yet powerful method. The idea behind it is good enough to make it compete in terms of quality of results with much more complicated methods using advanced techniques. [Open Text Summarizer, 2016] Still the fact that it is independent of the language of the text makes some space for improvements by adding heuristics without compromising its language independence (much). The main approaches to analyzing text are abstraction-based and extraction-based. The former analyzes the text and rephrases it by omitting details. This is what humans do when solving text summarization tasks. The latter identifies key sentences and selects them for inclusion in the summary, trying to keep the text coherent. A famous algorithm that uses this approach is Google's TextRank. Both OTS and our proposed algorithms are extraction-based.

Description of the proposed algorithm

We present a summary-generating algorithm based on OTS. Our aim is to improve the characteristics of the OTS, namely the quality of the generated summary and of the extracted keywords. At the same time we have tried to not complicate the algorithm with procedures that use "hard" NLP such as syntactic analysis, but by using some heuristics in order to include in the summary more semantically important

parts of the text. We propose a different way for identifying word-forms representative for the given text (and representative sentences based on that) by comparing the word frequency in the text to the word frequency in a large corpora.

The algorithm we propose makes use of the following additional to the text information:

stem of nouns, verbs, adjectives; abbreviations list for sentence segmentation; a stop-word list; a list of personal pronouns; a list of linking words and a list of substitutions for uniforming non-alphabetic characters.

Our algorithm consists of 10 main steps.

1. First, we make sentence segmentation. 2. Then we do string-replacements in two stages: a) We replace all non-traditional quotation marks with their most commonly used variant; b) we replace all short forms such as they've, doesn't, weren't, etc. with their full forms they have, does not, were not, etc. This is needed as the next step, 3., removes all the punctuation symbols and if the short forms are left unmodified, the part after the apostrophe will be left as a separate word and will not be recognized when compared against a language dictionary. 4. We stem all the word-forms. This is a common practice in many natural language processing tasks and is needed here because we are interested in the meaning, not the grammatical form of the word, as discussed in [Andonov et al. 2016]. 5. Stop-words are removed. These are short words such as conjunctions, pronouns, etc. that are frequent, but whose occurrence in the text does not give us information about its meaning. 6. After these preliminary steps, the score of every stemmed word in the text is calculated. The calculations use the frequency of the basic form of the word in an English corpora [Heuven et al. 2014] and the frequency in the text.

The formula for calculating the score is the following:

$$s_v = max(\frac{h_v}{max(h)} - \frac{g_v^{1.5}}{max(g)}, 0) \quad (1)$$

A default value is assigned to the words-forms in the text that do not exist in the corpora - the median of the corpora.

Thus, we obtain a higher score for those words that are more frequent in the text than in the corpora. Negative values for the score cannot be used so we replace them with zero, as given in (1). In the previous paper [Andonov et al. 2016], the heuristic formula we proposed was based on division of the two members given in (1). Our more extensive experiments after the first publication showed that it does not give satisfying results for words that have low frequencies in the corpora. As a result, now we apply subtraction, which has given much better results.

7. Once we have obtained the scores of individual words, we calculate the score of the sentences. In order to track the rule of the usual application of the semantic focal point in writing, here we use a heuristic that sentences in the beginning of the paragraph are slightly more important than those in the middle. Thus we give bigger weight to the first sentence of the text and to all first sentences in paragraphs. After that a link score is calculated, based on a list of expressions that we have created. The idea is that linking words and phrases such as 'for example', 'in addition', 'for instance', 'in particular', 'in fact', etc., especially when placed at the beginning of a sentence, mean that the sentence containing them is referencing to something said in the previous one. Thus if a sentence starts or has near its beginning such an expression then this score is higher. The reason for being interested in such sentences is that they should not be separated from the ones they are linking to, otherwise the coherence of the summary will deteriorate.

Our algorithm assigns different weights to different parts of speech.

From a linguistic point of view the subject and predicate are the most important. Without sentence structure analysis however the best we can do is to identify the objects and characters the sentence talks about. For that reason we consider nouns more important and verbs and adjectives less so.

$$S_s = pb_s * \sum_{i=1}^{|s|}(h_i * w_i) \qquad (2)$$

where S_s is the score of sentence s, pb_s is the paragraph beginning score of s, $|s|$ is the number of words in the sentence s, h_i is the score of every word i in the sentence s and w_i is the part of speech weight of every word i in sentence s.

8. At this step we mark the sentences for the summary. All sentences are sorted by their score and then the first one third with the highest scores are marked for inclusion. We use 1/3 of the original text as the selected length of the summary as this is a common practice when humans summarize text. However, this value can be changed. At this step, are also marked the sentences with high link score, as this indicates a high probability that a consequent sentence is referencing to content of the current one. We take this into account in order to improve the coherence of the generated summary.

9. After having extracted and assigned all this additional information to the text, the summary is created by adding all sentences marked for inclusion. Also the quotation marks of partially included in the summary quotes are fixed.

10. The final step is to list the words with the highest score as keywords.

Ideas for future development

In order to verify the hypothesis that our changes improve the quality of the summary we made a blind evaluation test by human experts of the summaries generated by both algorithms. One benefit of this is that the experts gave us not only their ratings to the summaries, but also valuable feedback. The classical understanding of a summary is 1/3 of the text, however when the text is large – for example for news and opinion articles, a summary of 5-7 pages is still pretty large and, for practical reasons, one expert pointed out that it would be better to make this fraction smaller. A similar remark was made about the number of keywords, as the fixed number of 5 keywords is inadequate for very small or very large articles.

After the analysis of the results we also came to some general conclusions concerning the strategy to be used in the future. From a data mining perspective, both algorithms do not use a potentially valuable bit of information – the topic of the article. The keywords and the summaries will probably be of better quality if a reliable entities detection algorithm is used, which is not the case at the moment. When the text is fairly long and/or consists of self-contained sections, our tests with manual splitting of the text and generating separate mini-summaries to be combined in a single one made us believe that summarizing the paragraphs one by one with local centers or utilizing a sliding window for the algorithm will be a better strategy.

Comparative analysis and some statistical parameters of the summaries

In order to evaluate the effect of the changes performed on the original algorithm, we did a comparative study using the following procedure:

The authors submitted the summaries followed by the full text of the articles/news items and the keywords to 4 evaluators. Each one had to read at least 7 summaries and keywords and evaluate their quality by reading beforehand the full texts. The experts had to use a scale from 1 to 5 where 1 is the worst and 5 is the best mark. The texts and what algorithm was used to generate the summary were chosen at random and the experts had no information about which summary is created by means of which algorithm.

After the evaluation, we performed a trivial procedure of scaling the assigned marks to the global mean mark as the evaluators had different requirements and different mean marks.

In order to study the influence of some parameters of the original text on the quality of the summaries, we measured the following characteristics of the original text:

1. NumberWords is the number of words in the text after segmentation

2. NumberParagraphs is the number of paragraphs in the text as detected by our segmentation procedure
3. Complexity – we are using the Automated Readability Index (ARI). This is a readability test designed to assess the understandability of a text. At the output ARI gives a number which approximates the grade level needed to comprehend the text. The formula it uses is shown below:

$$4.71 * (\frac{characters}{words}) + 0.5(\frac{words}{sentences}) - 21.43 \quad (3)$$

Richness – we measure the richness by using the Yule's *I* index

$$I = \frac{M_1^2}{M_2 - M_1} \quad (4)$$

where M_1 is the number of all word forms a text consists of and M_2 is the sum of the products of each observed frequency to the power of two and the number of word types observed with that frequency The larger Yule's *I*, the larger the diversity of the vocabulary (and thus, arguably, the more difficult the text) [Teller, 2011].

Table 1 gives the description of the measures characteristics of the sample of 50 papers and the marks, given by the evaluators (after correction).

Table 1. Descriptive statistics of the sample

	N	Minimum	Maximum	Mean	Std. Deviation
Number of Words	50	263	6706	978,94	1112,825
Number of Paragraphs	50	1	67	16,18	12,462
Complexity	50	7	20	13,03	3,033
Richness	50	6,20	42,72	20,7705	7,99629
Text Summary Mark	50	2,55	5,55	4,1224	0,73468
KeyWords Mark	35	1,86	5,29	3,8603	0,92854

Comparison of the results for the two summarizers.

Our relatively small sample has shown some promising tendencies. The comparative statistics are given in table 2.

Table 2. Comparative parameters of the evaluation of the two summarizers – OTS and the developed Improved OTS (IOTS)

Sumarizer		TextSunnaryMarkr	KeyWordsMark
IOTS	Mean	4,155	3,911428571
	N	22	14
	Std. Deviation	0,751	1,068520596
OTS	Mean	4,096	3,826190476
	N	28	21
	Std. Deviation	0,734	0,84884319
Total	Mean	4,122	3,860285714
	N	50	35
	Std. Deviation	0,734	0,928539288

As it is seen from the results in table 2, the mean mark for the quality of the text of the summary and the mean mark for the keywords extraction are better for the reported here Improved OTS (IOTS) summarizer reported here.

Analysis of the influence of the parameters of the text on the quality of the summary

In order to check whether the diffidence is statistically significant, we performed an independent sample T-test and ANOVA. Unfortunately, with such a small sample, having a small difference in the means and big variances of the evaluations, both tests cannot reject the hypothesis that the means of marks for the two summarizers are equal.

In order to investigate the possible means for further improvement, we examined the influence of the parameters of the original text on the quality of the generated summaries and the extraction of keywords (assuming that the experts' marks show this quality correctly).

One expects both algorithms explained in the previous section to be sensitive to the length of the text to be summarized. However, as shown in table 3, there is no effect of the number of words in the original text on the quality of the summary and the keywords.

Table 3. Correlation matrix (Pearson) – effect of the parameters of the text on the quality if the summary and on the keywords extraction

		TextSummary Mark	KeyWords Mark	Complexity	NumberWords	NumberParagraphs	Richness
TextSummary Mark	Correlation	*1*	-,196	-,297	,185	,168	-,256
	Sig. (2-tailed)		,260	,036	,199	,243	,073
	N	50	35	50	50	50	50
KeyWordsMark	Correlation	-,196	1	-,177	,170	,208	-,190
	Sig. (2-tailed)	,260		,308	,328	,231	,274
	N	35	35	35	35	35	35
Complexity	Pearson Correlation	-,297	-,177	1	,024	,037	,043
	Sig. (2-tailed)	,036	,308		,866	,798	,766
	N	50	35	50	50	50	50
NumberWords	Correlation	,185	,170	,024	1	,866	-,489
	Sig. (2-tailed)	,199	,328	,866		,000	,000
	N	50	35	50	50	50	50
NumberParagraphs	Correlation	,168	,208	,037	,866	1	-,527
	Sig. (2-tailed)	,243	,231	,798	,000		,000
	N	50	35	50	50	50	50
Richness	Correlation	-,256	-,190	,043	-,489	-,527	1
	Sig. (2-tailed)	,073	,274	,766	,000	,000	
	N	50	35	50	50	50	50

We observe another dependency – the quality of the summary is negatively correlated with the complexity of the original text (Parson Correlation -0.297, p-value 0.03). It seems the more the text is evaluated as complex (following equation (3), as explained in this section), the worse is the quality of the summary.

This dependency can be interpreted as follows: The term $\frac{words}{sentences}$ in ARI readability index (see equation (3)) expresses a measure of the length of an ‘’average” sentence in the text. As explained, the summarizers extract (leave in the summary) entire sentences, having calculated first scores concerning the words inside of each sentence, sentence by sentence. From where - the dependency we observe.

Conceder we have to express T simple thoughts in a text with N words. The ideal number of sentences would be T – one simple thought in one sentence. This is an ideal nonrealistic case in which the average length of a sentence would be N/T and the complexity (difficult readability) measure ARI would be small. Unfortunately for the summarizers of the considered type, the writing style contains complex subordinated, fused, concatenated etc. sentences, expressing more than one simple thought within a sentence. In result the text is judged as “complex” using expression (3), and the summarizers extract 1/3 not of T sentences, but of T-K longer complex sentences. That means: 1. each of the longer sentences is rated (see step 7 of the algorithm) using words which express more than one simple thought; 2. The summary contains less than T/3 longer sentences which risk being semantically unrelated, so the coherence of the summary is damaged.

Unfortunately, to solve this problem, one needs syntactic analysis to separate the sub-sentences, something that we want to avoid.

Analogical reasoning can be applied for the number of paragraphs.

The other path of reasoning in our analysis is related to the result concerning the dependency of the quality of the summary with the richness of the original text.

The formula (1) applied to calculate the scores does not take into consideration the use of synonyms. However, the writings of higher quality avoid the use of one and the same word several times, writers look for synonyms to make the text less repetitive and following expression (4) it becomes “richer”. Obviously that influences the frequencies used to calculate the scores and the quality of the summary. Once more time the problem touches semantic questions. However, this behavior of the algorithm can be adjusted using synonym dictionaries in a convenient way.

Conclusion

We have introduced several modifications in OTS that are showing promising results for improved quality of the summaries and keyword extraction. We found that the results are still not statistically significant and we did an analysis of the parameters of the original text and the quality of the summaries of *extraction-based* type. Our results show paths for further improvement of this type of summary generation.

Acknowledgment

This paper is published with partial support by the ITHEA ISS (www.ithea.org) and the Central Fund for Strategic development, New Bulgarian University.

We are grateful to the colleagues and friends who helped us with the summary evaluation. They are Galina Velichkova, Dessislava Petkova and Krassimir Todorov. We are grateful to our colleague Dimitar Atanasov for consulting our statistical approaches.

Bibliography

[Andonov et al. 2016] Filip Andonov, Velina Slavova, Marouane Soula. Heuristics-based classifier in a framework for sentiment analysis of news. International Journal "Information Content and Processing" Volume 3, Number 3, ITHEA, 2016. ISSN 2367-5128 (printed), ISSN 2367-5152 (online). pp. 224 - 234

[Heuven et al. 2014] Walter J. B. van Heuven, Pawel Mandera, Emmanuel Keuleers, and Marc Brysbaert, SUBTLEX-UK: *A new and improved word frequency database for British English*, The Quarterly Journal of Experimental Psychology Vol. 67, Iss. 6,2014

[Open Text Summarizer, 2016] Open Source Text Processing Project: Open Text Summarizer. http://textprocessing.org/2016/01, visited 08.2016

[Teller, 2011] Swizec Teller. Measuring vocabulary richness with python. September 28, 2011. https://swizec.com/blog/measuring-vocabulary-richness-with-python/swizec/2528, visited 08.2016

Authors' Information

Filip Andonov **-** *New Bulgarian University, department of Computer Science, fandonov@nbu.bg.*

Major Fields of Scientific *Research: multicriteria optimization, data mining, text processing, Python language*

Velina Slavova **-** *New Bulgarian University, department of Computer Science, vslavova@nbu.bg*

Major Fields of Scientific Research*: AI, Cognitive Science*

Georgi Petrov *- New Bulgarian University, department of Telecommunications, gpetrov@nbu.bg*

Major Fields of Scientific Research*: automation of processes*