

Terms Derived from Frequent Sequences for Extractive Text Summarization*

Yulia Ledeneva¹, Alexander Gelbukh¹, and René Arnulfo García-Hernández²

¹ Natural Language and Text Processing Laboratory,
Center for Computing Research, National Polytechnic Institute, DF 07738, Mexico
yledeneva@yahoo.com

www.Gelbukh.com

² Instituto Tecnológico de Toluca, Mexico
rearnulfo@hotmail.com

Abstract. Automatic text summarization helps the user to quickly understand large volumes of information. We present a language- and domain-independent statistical-based method for single-document extractive summarization, i.e., to produce a text summary by extracting some sentences from the given text. We show experimentally that words that are parts of bigrams that repeat more than once in the text are good terms to describe the text's contents, and so are also so-called maximal frequent sentences. We also show that the frequency of the term as term weight gives good results (while we only count the occurrences of a term in repeating bigrams).

1 Introduction

A summary of a document is a (much) shorter text that conveys the most important information from the source document. There are a number of scenarios where automatic construction of such summaries is useful. For example, an information retrieval system could present an automatically built summary in its list of retrieval results, for the user to quickly decide which documents are interesting and worth opening for a closer look—this is what Google models to some degree with the snippets shown in its search results. Other examples include automatic construction of summaries of news articles or email messages to be sent to mobile devices as SMS; summarization of information for government officials, businessmen, researchers, etc., and summarization of web pages to be shown on the screen of a mobile device, among many others.

The text summarization tasks can be classified into single-document and multi-document summarization. In single-document summarization, the summary of only one document is to be built, while in multi-document summarization the summary of a whole collection of documents (such as all today's news or all search results for a query) is built. While we believe that our ideas apply to either case, in this work we have experimented only with single-document summaries.

* Work done under partial support of Mexican Government (CONACyT, SNI, SIP-IPN, COTEPABE-IPN, COFAA-IPN). The authors thank Rada Mihalcea for useful discussion.

The summarization methods can be classified into abstractive and extractive summarization [1]. An abstractive summary is an arbitrary text that describes the contexts of the source document. Abstractive summarization process consists of “understanding” the original text and “re-telling” it in fewer words. Namely, an abstractive summarization method uses linguistic methods to examine and interpret the text and then to find new concepts and expressions to best describe it by generating a new shorter text that conveys the most important information from the original document. While this may seem the best way to construct a summary (and this is how human beings do it), in real-life setting immaturity of the corresponding linguistic technology for text analysis and generation currently renders such methods practically infeasible.

An extractive summary, in contrast, is a selection of sentences (or phrases, paragraphs, etc.) from the original text, usually presented to the user in the same order—i.e., a copy of the source text with most sentences omitted. An extractive summarization method only decides, for each sentence, whether or not it will be included in the summary. The resulting summary reads rather awkward; however, simplicity of the underlying statistical techniques makes extractive summarization an attractive, robust, language-independent alternative to more “intelligent” abstractive methods. In this paper, we consider extractive summarization.

A typical abstractive summarization method consists in several steps, at each of them different options can be chosen. We will assume that the units of selection are sentences (these could be, say, phrases or paragraphs). Thus final goal of the extractive summarization process is *sentence selection*. One of the ways to select the appropriate sentences is to assign some numerical measure of usefulness of a sentence for the summary and then select the best ones; the process of assigning these usefulness weights is called *sentence weighting*. One of the ways to estimate the usefulness of a sentence is to sum up usefulness weights of individual terms of which the sentence consists; the process of estimating the individual terms is called *term weighting*. For this, one should decide what the terms are: for example, they can be words; deciding what objects will count as terms is the task of *term selection*. Different extractive summarization methods can be characterized by how they perform these tasks.

The paper is organized as follows. Section 2 summarizes the state-of-the-art of text summarization methods. In Section 3, some notions used for term selection in our method are introduced. Section 4 presents our experimental setting. Sections 5 and 6 describe the obtained experimental results for different term selection and term weighting schemes, respectively, which are compared in Section 7 with those of existing methods. Section 8 concludes the paper.

2 Related Work

Ideally, a text summarization system should “understand” (analyze) the text and express its main contents by generating the text of the summary. For example, Cristea *et al.* [4] perform sentence weighting according to their proximity to the central idea of the text, which is determined by analysis of the discourse structure.

However, the techniques that try to analyze the structure of the text involve too sophisticated and expensive linguistic processing. In contrast, most of the methods discussed in the literature nowadays represent the text and its sentences as a bag of simple features, using statistical processing without any attempts to “understand” the text.

Supervised learning methods consider sentence selection as a classification task: they train a classifier using a collection of documents supplied with existing summaries. As features of a sentence such methods can consider text units (in which case we can speak of term selection) or other, non-lexical characteristics. Villatoro-Tello *et al.* [7] use as terms n -grams found in the text. Kupiec *et al.* [2] use predefined cue phrases (this makes the method language- and domain-dependent) as well as non-lexical features such as the position and length of the sentence; their sentence weighting procedure also includes measuring the overlap of the sentence with the title of the document. HaCohen-Kerner *et al.* [18] consider many other lexical and non-lexical features, such as the position of the sentence in the paragraph.

However, the majority of current methods are purely heuristic: they do not use any learning but directly state the procedure used for term selection, term weighting, and/or sentence weighting (given that sentence selection in most cases consists in selecting the best-weighted sentences).

A very old and very simple sentence weighting heuristic does not involve any terms at all: it assigns highest weight to the first sentences of the text. Texts of some genres—such as news reports or scientific papers—are specifically designed for this heuristic: e.g., any scientific paper contains a ready summary at the beginning. This gives a baseline [12] that proves to be very hard to beat on such texts. However, comparing term-based methods with such position-based baseline is not fair in the sense that this baseline only works on text of specific genres (say, it will not work on official documents, email messages, webpages, or literary novels) and uses information (the position of the sentence) not available to term-based methods. It is worth noting that in Document Understanding Conference (DUC) competitions [12] only five systems performed above this baseline, which does not demerit the other systems because this baseline is genre-specific. Though the method proposed in this paper very slightly outperforms this baseline, such a comparison is unfair.

Of the works devoted to term-based methods, most concentrate on term weighting. Xu *et al.* [6] derives relevance of a term from an ontology constructed with formal concept analysis. Song *et al.* [3] basically weight a word basing on the number of lexical connections, such as semantic associations expressed in a thesaurus, that the word has with its neighboring words; along with this, more frequent words are weighted higher. Mihalcea [15] presents a similar idea in the form of a neat, clear graph-based formalism: the words that have closer relationships with a greater number of “important” words become more important themselves, the importance being determined in a recursive way similar to the PageRank algorithm used by Google to weight webpages.

The latter idea can be applied directly to sentence weighting without term weighting: a sentence is important if it is related to many important sentences, where relatedness can be understood as, say, overlap of the lexical contents of the sentences [15]. The two methods presented in [15] are those that currently give the best results and with which we compare our suggested method.

While in the experiments reported in the papers discussed above were based on words as terms, this is not the only possible option. Liu *et al.* [5] uses pairs of syntactically connected words (basic elements) as atomic features (terms). Such pairs (which can be thought of as arcs in the syntactic dependency tree of the sentence) have been shown to be more precise semantic units than words [19, 20]. However, while we believe that trying text units larger than a word is a good idea, extracting the basic elements from the text requires dependency syntactic parsing, which is language-dependent. Simpler statistical methods (cf. the use of n -grams as terms in [7]) may prove to be more robust and language-independent.

In this paper we analyze several options for simple language-independent statistical term selection and corresponding term weighting, based on units larger than one word. In particular, we show that so-called maximal frequent sequences (MFSs), as well as single words that are part of bigrams repeated more than once in the text, are good terms to describe documents.

3 Frequent Sequences

An ngram is a sequence of n words. We say that an ngram occurs in a text if these words appear in the text in the same order immediately one after another. For example, a 4-gram (ngram of length 4) *words appear in the text* occurs once in the previous sentence, while *appear immediately after another* does not (these words do not appear on adjusting positions), neither does *the text appear in* (order is different).

The definition of ngram depends on what one considers words. For example, one can consider capitalized (*Mr. Smith*) and non-capitalized (*a smith*) words as the same word or as different words; one can consider words with the same morphological stem (*ask, asked, asking*), the same root (*derive, derivation*), or the same meaning (*occur, appear*) as the same word; one can omit the stop-words (*the, in*) when counting word positions, etc. Say, one can consider that in our example sentence above there occur the ngrams *we say* (capitalization ignored), *word appear* (plural ignored), *appear text* (*in the* ignored). This can affect counting the ngrams: if one considers *occur* and *appear* as equivalent and ignores the stop-words, then in our example sentence the bigram *appear text* occurs twice.

We call an ngram frequent (more accurately, β -frequent) if it occurs more than β times in the text, where β is a predefined threshold. Frequent ngrams—we will also call them frequent sequences (FSs)—often bear important semantic meaning: they can be multiword expressions (named entities: *The United States of America*, idioms: *kick the basket*) or otherwise refer to some idea important for the text (*the President's speech, to protest against the war*).

Our hypothesis is that FSs can express ideas both important and specific for the document. This can be argued in terms of *tf-idf* (term frequency—inverse document frequency, a notion well-known in information retrieval [21]): on the one hand, the idea expressed by an FS is important for the document if it repeatedly returns to it (high term frequency); on the other hand, the corresponding idea should be specific for this document, otherwise there would exist in the language a single word or at least an abbreviation to express it (high inverse document frequency). It is important

to note that this argument does not apply to 1-grams, i.e., single words. Therefore, we do not consider 1-grams as ngrams in the rest of this paper.

An ngram can be a part of another, longer ngram. All ngrams contained in an FS are also FSs. However, with the arguments given above one can derive that such smaller ngrams may not bear any important meaning by their own: e.g., *The United States of America* is a compound named entity, while *The United* or *States of America* are not. Exceptions like *The United States* should not affect much our reasoning since they tend to be synonymous to the longer expression, and the author of the document would choose one or another way to refer to the entity, so they should not appear frequently both in the same document.

FSs that are not parts of any other FS are called Maximal Frequent Sequences (MFSs) [10, 11]. For example, in the following text

... *Mona Lisa* is the most beautiful picture of *Leonardo da Vinci* ...
 ... *Eiffel tower* is the most beautiful tower ...
 ... *St. Petersburg* is the most beautiful city of *Russia* ...
 ... The most beautiful church is not located in *Europe* ...

the only MFS with $\beta = 3$ is *is the most beautiful*, while the only MFS $\beta = 4$ is *the most beautiful* (it is not an MFS with $\beta = 3$ since it is not maximal with this β). As this example shows, the sets of MFSs with different thresholds do not have to, say, contain one another.

One of our hypotheses was that only MFSs should be considered as bearing important meaning, while non-maximal FSs (those that are parts of another FS) should not be considered. Our additional motivation was cost vs. benefit considerations: there are too many non-maximal FSs while their probability to bear important meaning is lower. In any case, MFSs represent all FSs in a compact way: all FSs can be obtained from all MFSs by bursting each MFS into a set of all its subsequences. García [10] proposed an efficient algorithm to find all MFSs in a text, which we also used to efficiently obtain and store all FSs of the document.

The notions of FSs and MFSs are closely related to that of repeating bigrams; see Section 5. This set is conceptually simpler, but for computational implementation MFSs could be more compact.

4 Experimental Setting

We have conducted several experiments to verify our hypotheses formulated in the previous section.

Algorithm. In each experiment, we followed the standard sequence of steps:

- Term selection: decide which features are to be used to describe the sentences;
- Term weighting: decide how the importance of each feature is to be calculated;
- Sentence weighting: decide how the importance of the features is to be combined into the importance measure of the sentence;
- Sentence selection: decide which sentences are selected for the summary.

The specific settings for each step varied between the experiments and are explained below for each experiment.

Test data set. We used the DUC collection provided [12]. In particular, we used the data set of 567 news articles of different length and with different topics. Each document in the DUC collection is supplied with a set of human-generated summaries provided by two different experts.¹ While each expert was asked to generate summaries of different length, we used only the 100-word variants.

Evaluation procedure. We used the ROUGE evaluation toolkit [13] which was found to highly correlate with human judgments [14]. It compares the summaries generated by the program with the human-generated (gold standard) summaries. For comparison, it uses n -gram statistics. Our evaluation was done using n -gram (1, 1) setting of ROUGE, which was found to have the highest correlation with human judgments, namely, at a confidence level of 95%.

5 Term Selection

For term selection, we compared MFSs with more traditional features such as single words and ngrams. Namely, we considered the following variants of term selection:

- M : the set of all MFSs, i.e., an ngram $m \in M$ if it is an MFS with some threshold β (recall that MFSs are of 2 words or longer and $\beta \geq 2$)² In the example from Section 3, $M = \{is\ the\ most\ beautiful,\ the\ most\ beautiful\}$. Also, we denote by M_2 the set of all MFSs with $\beta = 2$.
- B : repeating bigrams, i.e., bigrams with frequency at least 2. It is easy to show that it is the same set as the set of all bigrams from MFSs: a bigram $b \in B$ iff there exists an MFS $m \in M$ such that $b \subseteq m$. What is more, considering in the latter definition M_2 instead of M also gives the same set. In our example, $B = \{is\ the,\ the\ most,\ most\ beautiful\}$.
- W : single words (unigrams) from elements of B or, which is the same, of M . Namely, a word $w \in W$ if there exists a bigram $b \in B$ such that $w \in b$; it is easy to show that $w \in W$ iff there exists an MFS $m \in M$ such that $w \in m$. Again, considering M_2 in the latter definition also gives the same set. In our example, $B = \{is,\ the,\ most,\ beautiful\}$.

We give different definitions of the sets B and W to show that they are naturally derived from the notion of MFS and at the same time can be efficiently calculated.

Optionally, stop-words were eliminated at the pre-processing stage; in this case our bigrams (or MFSs) could span more words in the original text, as explained in Section 3.

For term weighting, the frequency of the term was used; for sentence weighting, the sum of the weights of the terms contained in the sentence was used; for sentence

¹ While the experts were supposed to provide extractive summaries, we observed that the summaries provided in the collection were not strictly extractive: the experts considerably changed the sentences as compared with the original text.

² In practice, we only considered the MFSs with the thresholds $\beta = 2, 3,$ and 4 , since MFSs with higher thresholds were very rare in our collection, except for those generated by stop-words.

selection, the sentences with greater weight were selected until the desired size of the summary (100 words) is reached.

Table 1 shows the results. Since the size of all summaries is the same (100 words), either measure (recall or precision) can be used for comparison.

Table 1. Recall on 100-words summaries for different term selection options

Terms	With stop-words	Without stop-words
<i>W</i> : words from <i>B</i> or <i>M</i>	0.39421	0.41371
<i>B</i> : repeating bigrams	0.40810	0.42173
<i>M</i> : all MFSs	0.43066	0.44085

As a kind of statistical significance check, we randomly divided our test data into two halves and ran this (and most of the other) experiments separately on each subset. These experiments confirmed the qualitative observations reported in this paper.

As Table 1 shows, MFSs are a promising choice for term selection. This motivated our further experiments with term selection schemes derived from them, as well as with term weighting options for them.

6 Term Weighting and Sentence Selection

Inspired by the above results, we further experimented with MFSs and other term selection options derived from them. In addition to *M* and *W* from Section 5, we considered a generalization of the sets considered in Section 5:

- *N*: all ngrams from MFSs, i.e., an ngram $n \in N$ if there exists an MFS $m \in M$ such that $n \subseteq m$ (including single words, i.e., 1-grams). Again, considering in the latter definition M_2 also gives the same set, which allows for efficient calculation of the set *N* in practice. In our example, $N = \{is, the, most, beautiful, is\ the, the\ most, most\ beautiful, is\ the\ most, the\ most\ beautiful, is\ the\ most\ beautiful\}$. Note that $W \subset N, M \subset N$.
- $N \setminus W, N \setminus M_2, N \setminus (W \cup M_2)$: same as *N* but not including 1-grams, the whole MFS, or both; here M_2 is the set of MFSs with $\beta = 2$. In our example, $N \setminus (W \cup M_2) = \{is\ the, the\ most, most\ beautiful, is\ the\ most, the\ most\ beautiful\}$.

Optionally, stop-words were eliminated at the pre-processing stage. For term weighting, different formulae were considered containing the following values:

- *f*: frequency of the term in MFSs, i.e., the number of times the term occurs in the text within some MFS. In our example, $f(is) = 3$ since it occurs 3 times in the text within the MFS *is the most beautiful*. If the term itself is an MFS, then this is just the frequency of this term in the text (e.g., for *M*, *f* is the same as term weight in Section 5; for *W* and *N* it is not). Under certain realistic conditions (MFSs do not intersect in the text, words do not repeat within one MFS) *f* is the number of times the term occurs in the text as part of a repeating bigram. In our example, $f(is) = 3$ since it occurs 3 times in a repeating bigram *is the* (and one time in a non-repeating context *church is not*).

- l : the maximum length of an MFS containing the term. In our example, $l(is) = 4$ since it is contained in a 4-word MFS *is the most beautiful*.
- 1: the same weight for all terms.

For sentence weighting, the sum of the weights of the terms contained in the sentence was used. For sentence selection, the following options were considered:

- best: sentences with greater weight were selected until the desired size of the summary (100 words) is reached. This is the most standard method.
- k best+first: k best sentences were selected, and then the first sentences of the text weight were selected until the desired size of the summary is reached. This was motivated by the very hard-to-beat baseline mentioned in Section 2: only the very best sentences according to our weighting scheme might prove to be above this baseline.

The results are shown in Table 2. We conducted our experiments in three phases. From Table 1 we knew that term selection scheme M with stop-words removed gave the best results with other parameters fixed (term weighting, sentence weighting, and sentence selection). So we started from modifying these parameters for the same term selection scheme; see the upper third part of Table 2. The first line of the table represents the best result from Table 1. The best results are highlighted in boldface. From this experiment, we discarded the term weighting options related to l .

Table 2. Results for different term selection options

Term Selection		Term weight- ing	Sentence Selection	Results		
Terms	Stop- words			Recall	Precision	F-measure
M	excluded	f	best	0.44085	0.45564	0.44796
		1		0.44128	0.45609	0.44840
		l		0.43977	0.45587	0.44752
		l^2		0.42995	0.44766	0.43847
				0.43812	0.45411	0.44581
	included	$l \times f$		0.43353	0.44737	0.44022
W	included	f	best	0.44582	0.45820	0.45181
	excluded	1		0.44609	0.45953	0.45259
		f^2		0.38364	0.40277	0.39284
				0.43892	0.45265	0.44556
N		f or 1	0.43711	0.45099	0.44383	
W	excluded	f	1best+first	0.46576	0.48278	0.47399
			2best+first	0.46158	0.47682	0.46895
1		1best+first	0.46354	0.48072	0.47185	
		2best+first	0.46028	0.47567	0.46772	
M		l	1best+first	0.46381	0.48124	0.47223
			2best+first	0.45790	0.47430	0.46583

Then we tried other term selection options, such as W and N , with the term weighting option 1 and the options related to f , which showed good performance in the first experiment. The results are shown in the middle third of Table 2. Term selection W gave a slightly better result than M . The results for N are equal with f and 1 as weighting. Other combinations based on N did not give good results; see Table 3 (stop-words excluded, *best* sentence selection).

Finally, with the best combinations obtained from the first two experiments, we tried different sentence selection variants; see the last third of Table 2.

One can observe that any k best+first sentence selection option outperformed any combination that used the standard sentence selection scheme, with smaller k always giving better results—that is, only the slightest correction to the baseline improved it. The best result was obtained with single words derived from MFSs, with their weighting by the frequency of the corresponding MFS.

Table 3. Results for variants of the set N (options: *excluded, best*)

Terms	Term weighting	Recall	Precision	F-measure
N	f or 1	0.43711	0.45099	0.44383
	l	0.42911	0.44324	0.43594
$N \setminus W$	1	0.42009	0.43693	0.42823
	f	0.41849	0.43532	0.42662
$N \setminus M_2$	1	0.42315	0.43806	0.43035
$N \setminus (W \cup M_2)$		0.41084	0.42759	0.41893

7 Comparison

We compared the following results:

- State of the art: The author of [15] provided us with her data, which were evaluated in the same conditions as proposed methods. Specifically, DirectedBackward version of TextRank [15] was evaluated. We also list the results of the original TextRank with implementation of PageRank with DirectedBackward version of TextRank but with some additional data processing to remove noisy data [16] and the modified TextRank with a biased version of PageRank [17]. See details of the preprocessing in [15–17].
- Baseline: We denote *Baseline: first* the baseline mentioned in Section 2, which selects the first sentences of the text until the desired size of the summary is reached [12]. This baseline gives very good results on the kind of texts (news reports) that we experimented with, but would not give so good results on other types of texts. Thus we proposed another baseline, denoted *Baseline: random*, which selects random sentences; the results presented below are averaged by 10 runs. We believe this to be a more realistic baseline for the types of texts other than news reports.

- Our proposal: We compare these methods with the best results obtained with our proposal with the *best* and *1best+first* sentence selection scheme, as shown in Table 2. In both cases our best results were obtained with the options *W* without stop-words for term selection and *f* for term weighting.

For fair comparison, we separated the methods by the type of information they used in addition to the weighting derived from terms:

- None (text is considered as a bag of sentences, sentence as a bag of terms, terms as strings),
- Order of sentences (say, first sentences are treated specially),
- Sophisticated pre-processing to obtain the terms.

We believe that in the future combination of these types of additional information can give even better results. The comparison is given in Table 4.

Table 4. Results with other methods

Additional info used	Method	Recall	Precision	F
None	Baseline: <i>random</i>	0.37892	0.39816	0.38817
	TextRank: [15]	0.45220	0.43487	0.44320
	Proposed: <i>W, f, best</i>	0.44609	0.45953	0.45259
Order of sentences	Baseline: <i>first</i>	0.46407	0.48240	0.47294
	Proposed: <i>W, f, 1best+first</i>	0.46576	0.48278	0.47399
Pre-processing	TextRank: [16]	0.46582	0.48382	0.47450
	TextRank: [17]	0.47207	0.48990	0.48068

We could not apply our method with the pre-processing option because we did not have access to the specific details of the pre-processing procedure used in [16] and [17]. However, in the other two categories our method outperformed the others. Possibly with the same type of pre-processing our method would outperform the others in the last category, too.

8 Discussion and Conclusions

We have tested different combinations of term selection, term weighting, and sentence selection options for language- and domain-independent extractive single-document text summarization on a news report collection.

We observed that words from repeating bigrams are good terms, and so are MFSs (we can speculate that MFSs are still better semantic units but splitting them into single words gives a more flexible and less sparse comparison). For term weighting, we observed that a good weighting scheme is the number of occurrences of the term in the text as part of a repeating bigram. With these settings, we obtained the results superior to the existing state-of-the-art methods.

Most of the state-of-the-art methods perform worse than the baseline method that takes into account a special ordering of sentences in news reports, which contain a nearly ready abstract in their first sentences. However, our methods can select one

sentence better than this baseline method (while already the second-best sentence selected by our method proves to be worse than the baseline). This gives a hybrid method (one sentence our and then back-off to the baseline) superior to both the baseline and other state-of-the-art methods.

In our experiments we did not apply pre-processing that was shown to be beneficial for other methods, so our results are below those of other methods when they do apply it, though above them when they do not. The latter makes us believe that when we apply pre-processing we will obtain results superior to all existing methods. This will be the topic of our future work.

On the other hand, our experiments show that very different options (some of them rather absurd) only slightly affect the overall result, at least on the collection we used for our experiments. This can probably be explained by the nature of the texts in this collection (short news reports) and maybe by the behavior of the ROUGE evaluation scheme: the completely random selection baseline is rather high (so nearly any method would give at least similar results) while what seems to be almost top-line—selecting the first sentences of the text³—is quite low and quite near to the random baseline. This makes us rather pessimistic about much further progress in the results unless another data collection is used and probably better evaluation schemes are developed.

References

1. Lin, C.Y., Hovy, E.: Automated Text Summarization in SUMMARIST. In: Proc. of ACL Workshop on Intelligent, Scalable Text Summarization, Madrid, Spain (1997)
2. Kupiec, J., Pedersen, J.O., Chen, F.: A Trainable Document Summarizer. In: Proceedings of the 18th ACM-SIGIR Conference on Research and Development in Information Retrieval, Seattle, pp. 68–73 (1995)
3. Song, Y., et al.: A Term Weighting Method based on Lexical Chain for Automatic Summarization. In: Gelbukh, A. (ed.) CICLing 2006. LNCS, vol. 3878, Springer, Heidelberg (2006)
4. Cristea, D., et al.: Summarization through Discourse Structure. In: Gelbukh, A. (ed.) CICLing 2006. LNCS, vol. 3878, Springer, Heidelberg (2006)
5. Liu, D., et al.: Multi-Document Summarization Based on BE-Vector Clustering. In: Gelbukh, A. (ed.) CICLing 2006. LNCS, vol. 3878, Springer, Heidelberg (2006)
6. Xu, W., Li, W., et al.: Deriving Event Relevance from the Ontology Constructed with Formal Concept Analysis. In: Gelbukh, A. (ed.) CICLing 2006. LNCS, vol. 3878, Springer, Heidelberg (2006)
7. Villatoro-Tello, E., Villaseñor-Pineda, L., Montes-y-Gómez, M.: Using Word Sequences for Text Summarization. In: Sojka, P., Kopeček, I., Pala, K. (eds.) TSD 2006. LNCS (LNAI), vol. 4188, pp. 293–300. Springer, Heidelberg (2006)
8. Chuang, T.W., Yang, J.: Text Summarization by Sentence Segment Extraction Using Machine Learning Algorithms. In: Proc. of the ACL 2004 Workshop, Barcelona, España (2004)
9. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. *Information processing & Management* 24, 513–523 (1988)

³ We believe this to be nearly top-line because the first lines of a news report are intended by its author to serve as a ready summary of the whole report, and are probably the best summary for it.

10. García-Hernández, R.A., Martínez-Trinidad, J.F., Carrasco-Ochoa, J.A.: A Fast Algorithm to Find All the Maximal Frequent Sequences in a Text. In: Sanfeliu, A., Martínez Trinidad, J.F., Carrasco Ochoa, J.A. (eds.) CIARP 2004. LNCS, vol. 3287, pp. 478–486. Springer, Heidelberg (2004)
11. García-Hernández, R.A., Martínez-Trinidad, J.F., Carrasco-Ochoa, J.A.: A New Algorithm for Fast Discovery of Maximal Sequential Patterns in a Document Collection. In: Gelbukh, A. (ed.) CICLing 2006. LNCS, vol. 3878, pp. 514–523. Springer, Heidelberg (2006)
12. DUC. Document understanding conference 2002 (2002), <http://www-nlpir.nist.gov/projects/duc>
13. Lin, C.Y.: ROUGE: A Package for Automatic Evaluation of Summaries. In: Proceedings of Workshop on Text Summarization of ACL, Spain (2004)
14. Lin, C.Y., Hovy, E.: Automatic Evaluation of Summaries Using N-gram Co-Occurrence Statistics. In: Proceedings of HLT-NAACL, Canada (2003)
15. Mihalcea, R.: Random Walks on Text Structures. In: Gelbukh, A. (ed.) CICLing 2006. LNCS, vol. 3878, pp. 249–262. Springer, Heidelberg (2006)
16. Mihalcea, R., Tarau, P.: TextRank: Bringing Order into Texts. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2004), Barcelona, Spain (2004)
17. Hassan, S., Mihalcea, R., Banea, C.: Random-Walk Term Weighting for Improved Text Classification. In: Proceedings of the IEEE International Conference on Semantic Computing (ICSC 2007), Irvine, CA (2007)
18. HaCohen-Kerner, Y., Zuriel, G., Asaf, M.: Automatic Extraction and Learning of Keyphrases from Scientific Articles. In: Gelbukh, A. (ed.) CICLing 2006. LNCS, vol. 3878, pp. 645–657. Springer, Heidelberg (2006)
19. Bolshakov, I.A.: Getting One’s First Million...Collocations. In: Gelbukh, A. (ed.) CICLing 2004. LNCS, vol. 2945, pp. 229–242. Springer, Heidelberg (2004)
20. Koster, C.H.A.: Transducing Text to Multiword Units. In: Workshop on Multiword Units MEMURA at the fourth International Conference on Language Resources and Evaluation, LREC-2004, Lisbon, Portugal (2004)
21. Baeza Yates, R., Ribeiro Neto, B.: Modern Information Retrieval. ACM Press, New York (1999)