Text Summarization by Sentence Extraction Using Unsupervised Learning*

René Arnulfo García-Hernández, Romyna Montiel, Yulia Ledeneva,
Eréndira Rendón, Alexander Gelbukh, and Rafael Cruz

Pattern Recognition Laboratory, Toluca Institute of Technology, Mexico
Autonomous University of the State of Mexico, Mexico
Center for Computing Research, National Polytechnic Institute, Mexico
SoNet RC, University of Center Europe in Skalica, Slovakia
renearnulfo@hotmail.com, romyna.montiel@yahoo.com.mx,
yledeneva@yahoo.com, www.Gelbukh.com

Abstract. The main problem for generating an extractive automatic text summary is to detect the most relevant information in the source document. Although, some approaches claim being domain and language independent, they use high dependence knowledge like key-phrases or golden samples for machine-learning approaches. In this work, we propose a language- and domain-independent automatic text summarization approach by sentence extraction using an unsupervised learning algorithm. Our hypothesis is that an unsupervised algorithm can help for clustering similar ideas (sentences). Then, for composing the summary, the most representative sentence is selected from each cluster. Several experiments in the standard DUC-2002 collection show that the proposed method obtains more favorable results than other approaches.

1 Introduction

In the last two decades, we have experienced an exponential increase in the electronic text information available for being query. The best example of the hugest and ever-increased collection of documents most frequently querying is Internet, with millions of web documents. Nowadays, it is common to use Google for retrieving a list of thousands of web pages, but the user has to decide if a document is interesting only with the extracted text where the words of the request query appears. Therefore, it is necessary download and read each document until the user finds satisfactory information. It was unnecessary and time-consuming routine. Thus, it is indispensable to develop automatic methods for detecting the most relevant content from a source document in order to show it as a summary. In addition, there are a number of scenarios where automatic construction of such summaries is useful. Other examples include automatic construction of summaries of news articles or email messages for sending them to mobile devices as SMS; summarization of information for government officials, businesspersons, researches, etc., and summarization of web pages to

* Work done under partial support of Mexican Government: CONACyT, SNI, SIP-IPN, PIFI-IPN.

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 133–143, 2008.

be shown on the screen of a mobile device, among many others. These examples show that it is desirable that text summarization approaches work more in language and dominion independent way.

Automatic Text Summarization (ATS) is an active research area that deals with single- and multi-document summarization tasks. In single-document summarization, the summary of only one document is 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.

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 composed with 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 extractive 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.

The main problem for generating an extractive automatic text summary is to detect the most relevant information in the source document. Although, some approaches claim being domain and language independent, they use some degree of language knowledge like lexical information [2], key-phrases [3] or golden samples for supervised learning approaches [4-6]. Furthermore, training on a specific domain tends to customize the extraction process to that domain, so the resulting classifier is not necessarily portable. In our opinion, these works present a high dominion and language dependence degree.

In this work, we propose a language- and domain-independent automatic text summarization approach by sentence extraction using an unsupervised learning algorithm. Our hypothesis is that an unsupervised algorithm can help for clustering similar ideas (sentences). Then, for composing the summary, the most representative sentence is selected from each cluster. In addition, this approach lets to control in some degree the number of words in the generated summary.

The paper is organized as follows. Section 2 summarizes the state-of-the-art of text summarization methods. Section 3 describes the general scheme of the proposed approach. Section 4 presents the experimental settings followed for the experimentation. Section 5 compares our approach with those of existing methods. Section 6 concludes the paper.

2 Related Work

Ledeneva *et al.* [7, 8] suggest a typical automatic extractive summarization approach composed by term selection, term weighting, sentence weighting and sentence selection steps. 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.

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.* [9] 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 classification: 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.* [4] use as terms *n*-grams found in the text. Kupiec *et al.* [10] 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.* [3] 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 [11] that proves to be very hard to beat on such texts. It is worth noting that in Document Understanding Conference (DUC) competitions [11] only five systems performed above this baseline, which does not demerit the other systems because this baseline is genre-specific.

Of the works devoted to term-based methods, most concentrate on term weighting. Xu *et al.* [12] derives relevance of a term from an ontology constructed with formal concept analysis. Song *et al.* [2] 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 [13, 14] 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 web pages.

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 [13].

Recently, a novel approach quite different from other methods was presented by Ledeneva *et al.* [7]. In this work, the sentences are weighted by using the terms derived from the maximal frequent word sequences. Then, the best sentence is combined with the baseline sentences for composing the summary. This approach is ranked, according to ROGUE evaluation system, in third place.

The methods presented in [13-15] and [7] 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.* [16] 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 [17, 18, 19]. 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, as the use of *n*-grams as terms in [4], may prove to be more robust and language-independent.

3 General Scheme of the Proposed Approach

Usually an extractive summarization approach performs term selection, term weighting, sentence weighting, and sentence selection steps. However, the strategy of sentence selection step is reduced to simply taking the sentences with highest weight. Even though this strategy works well for the first ranked sentence, the strategy could lead for sentences similar to the first one to tend to be ranked after the first one; producing redundant sentences in the summary. This problem affects negatively in recall measure. In this sense, we propose to substitute the sentence weighting and sentence selection steps, with an unsupervised learning algorithm. Our hypothesis is that an unsupervised learning algorithm could help for automatically detecting the groups of similar sentences from which is selected the most representative sentence; reducing in this way the redundancy in the summary. In this section, we describe the general steps that are followed in the proposed approach.

3.1 Term Selection

An *n*-gram is a sequence of *n* words. We say that an *n*-gram occurs in a text if these words appear in the text in the same order immediately one after another, e.g., a 4-gram (*n*-gram 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 n-gram 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 *n*-grams *we say* (capitalization ignored), *word appear* (plural ignored), *appear text* (*in the* ignored). This can affect counting the *n*-grams: if one considers *occur* and *appear* as equivalent and ignores the stop-words, then in our example sentence the bigram *appear text* occurs twice.

3.2 Term Weighting

Boolean Weighting (BOOL): It is the easiest way to weight a term. It models the presence or absence of a term in the document, defined as:

$$w_i(t_j) = \begin{cases} 1, & \text{if the term } t_j \text{ appears in document } i \\ 0, & \text{otherwise} \end{cases}$$

Term Frequency (TF) was proposed in [20]. This weighting takes into account that a term that occurs in a document can better reflect the contents of document than a term that occurs less frequent. Therefore, the weighting TF assigns a greater relevance to terms with greater frequency and consists in evaluating the number of times the term appears in the document.

$$w_i(t_j) = f_{ij} \text{ , where } f_{ij} \text{ is the frequency of the term } j \text{ in document } i.$$

Inverse Document Frequency (IDF) was proposed by Salton [21] for improving information retrieval systems (IR). The problem of TF weighting in IR is that, when a term appears in almost all the documents in the collection; this term is useless for discriminating relevant documents. For example, the stop-word *and* could have a high TF, but it is useless for discriminating the relevant documents since tends to appear in most of the documents. IDF is defined as:

$$w_i(t_j) = \log\left(\frac{N}{n_j}\right),$$

where N is the number of documents in the collection and n_j is the number of documents where the term j appears.

TF-IDF. The problem of IDF weighting in IR is that it is not possible distinguish between two documents with the same vocabulary (list of different words), even thought if the term is more frequent in a document. TF-IDF weighting gives more relevance to the terms that are less frequent in the collection but more frequent into the document.

$$w_i(t_j) = f_{ij} \times \log\left(\frac{N}{n_j}\right)$$

Note that in this paper we propose to use these term weights for single document summarization. Therefore, for applying these term weights we can consider the document as a collection of sentences instead of a collection of documents.

3.3 Sentence Selection Using an Unsupervised Learning

An unsupervised learning algorithm form groups of objects in order to achieve, in the one hand, the greatest possible similarity between objects of a group, in other hand, the greatest possible dissimilarity between objects of different groups.

In this step, we propose to use an unsupervised algorithm for discovering the groups of sentences with similar meaning. Then, we can select the most representative sentence from each group in order to compose the summary. In particular, we propose to use the well-known K-means algorithm, which assumes that the number of clusters is previously known. Sometimes this characteristic is a disadvantage in K-means, however in our proposed approach is an advantage because let to specify the number of groups to create what allowed, at the same time, to estimate the number of words in the final summary. For example, if the average of words per sentence is 20 and a user desires a 100-word summary then K-means must create 5 clusters, obviously this is only an estimation of the number of words in the final summary. K-means represents each sentence in a vector space model. So, each document is represented as a vector of features, where the features correspond to the different terms in the document, in this case *n*-grams.

K-means is based on centroids, which are points in the vector space model calculated as the mean of the objects in a group. K-means iteration consists in to assign each object to the closest centroid and then the new centroids are recalculated again. The algorithm finishes when the centroids do not change. In the beginning, the K-means algorithm need seeds as the initial centroids for each group. Thus, the successful of K-means depends on selecting good initial seeds. Normally, the initial seeds are selected in random way. In our proposed approach, first sentences are considered as initial seeds, since it is known that the Baseline sentences are good candidate sentences for composing the summary. For measuring the similarity between two sentences the Euclidean distance is used, defined as:

$$\text{Distance}(X,Y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \ ,$$

where *X* and *Y* are sentences expressed as vectors with *n* features.

4 Experimental Results

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:

- *Preprocessing*: eliminate stop-words, then apply Porter stemming [24];
- *Term selection*: decide which size of *n*-grams as features are to be used to describe the sentences;
- *Term weighting*: decide how the importance of each feature is to be calculated, it can be BOOL, TF, IDF or TFIDF;
- *Sentence clustering*: decide the initial seeds for the K-means algorithm, in this case Baseline sentences;
- *Sentence selection*: after K-means finishes, select the closest sentence (most representative sentence) to each centroid for composing 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 standard DUC 2002 collection [11]. 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 [22] which was found to highly correlate with human judgments [23]. 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%. ROUGE calculates Precision, Recall, and F-measure values. We consider F-measure because it represents a balance (not an average) of Recall and Precision results.

Next tables show the results obtained with ROUGE for different gram size and different term weights. In the beginning, we only test for 1-grams to 5-grams, however, the results tends to be better as the size of *n*-gram was increased. Therefore, it was necessary to test until 11-gram. In particular, tables 1, 2 and 3 shows Recall, Precision and F-measure results, respectively. As F-measure shows, the worst result was obtained with 1-grams and BOOL; and the best result was obtained with 10-grams and IDF. In addition, these combinations were the worst and best results for recall and precision.

[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.

Table 1. Recall on 100-word summaries for different sizes of *n*-grams and its weights

Term Selection	Term Weighting			
	BOOL	TF	IDF	TFIDF
1-grams	0.47517	0.47686	0.47632	0.47545
2-grams	0.47705	0.47694	0.47779	0.47777
3-grams	0.47940	0.47940	0.47932	0.47932
4-grams	0.47891	0.47891	0.47916	0.47913
5-grams	0.47942	0.47942	0.47910	0.47910
6-grams	0.47989	0.47979	0.48020	0.48020
7-grams	0.47976	0.47992	0.47964	0.47993
8-grams	0.48113	0.48072	0.48075	0.48055
9-grams	0.48084	0.48084	0.48020	0.48109
10-grams	0.48058	0.48103	**0.48155**	0.48101
11-grams	0.48004	0.47903	0.47856	0.47856

Table 2. Precision 100-word summaries for different sizes of *n*-grams and its weights

Term Selection	Term Weighting			
	BOOL	TF	IDF	TFIDF
1-grams	0.47039	0.47219	0.47168	0.47078
2-grams	0.47211	0.47204	0.47284	0.47284
3-grams	0.47452	0.47454	0.47441	0.47441
4-grams	0.47410	0.47410	0.47432	0.47429
5-grams	0.47462	0.47462	0.47432	0.47432
6-grams	0.47495	0.47497	0.47530	0.47530
7-grams	0.47493	0.47510	0.47487	0.47512
8-grams	0.47633	0.47606	0.47588	0.47587
9-grams	0.47632	0.47632	0.47553	0.47654
10-grams	0.47575	0.47609	**0.47684**	0.47634
11-grams	0.47529	0.47409	0.47370	0.47370

Table 3. F-measure on 100-word summaries for different sizes of *n*-grams and its weights

Term Selection	Term Weighting			
	BOOL	TF	IDF	TFIDF
1-grams	0.47264	0.47439	0.47387	0.47298
2-grams	0.47445	0.47436	0.47519	0.47517
3-grams	0.47683	0.47684	0.47673	0.47673
4-grams	0.47638	0.47638	0.47661	0.47658
5-grams	0.47689	0.47689	0.47658	0.47658
6-grams	0.47729	0.47725	0.47762	0.47762
7-grams	0.47721	0.47738	0.47713	0.47739
8-grams	0.47860	0.47826	0.47818	0.47808
9-grams	0.47845	0.47845	0.47773	0.47868
10-grams	0.47803	0.47842	**0.47906**	0.47854
11-grams	0.47753	0.47642	0.47599	0.47599

5 Comparison

We compared our proposed approach with the following results:

- State of the art 1: The author of [13] provided data, which were evaluated in the same conditions as proposed methods. Specifically, DirectedBackward version of TextRank [13] 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 [14] and the modified TextRank with a biased version of PageRank [15]. See details of the preprocessing in [13–15].
- State of the art 2: We compare our methods with the best results reported in [7].
- Baseline (first): We denote *Baseline: first* the baseline mentioned in Section 2, which selects the first sentences of the document until the desired size of the summary is reached [7]. This Baseline gives very good results on the kind of documents (news reports) that we experimented with, but would not give so good results on other types of texts.
- Baseline (random): Ledeneva *et al.* [7] proposed another baseline heuristic, denoted *Baseline: random*, that is the average evaluation of selecting 10-times random sentences; we believe this to be a more realistic baseline for the types of texts other than news reports.

The comparison of the best F-measure results of our proposed approach with the above state-of-the-art approaches is presented in table 4. In this table, it is possible to observe that the difference between the worst and best approach is 0.09251. It difference was calculate in order to show that a centesimal or millesimal increase in F-measure is significant. The difference between Baseline (*first)* and Ledeneva [11] is 0.00105; and between Ledeneva [11] and TextRank [14] is 0.00051; however, the difference between TextRank [14] and our proposed approach is 0.00456. In addition, we calculate the difference between recall and precision in order to show how our proposed method gets the best balance between recall and precision. In addition, it is interesting to observe that, in comparison with Baseline (first), our approach obtained better recall results in all the experiments, while the precision was worse in all the experiments. Nevertheless, our proposed approach obtains better F-measure results than Baseline (first), except for 1-gram and BOOL weighting. It is good to mention the best recall result was obtained by our proposed approach.

Table 4. Results with other methods

Method	Recall	Precision	F-measure	|Recall-Precision|
Baseline: *random*	0.37892	0.39816	0.38817	0.01924
TextRank: [13]	0.45220	0.43487	0.44320	0.01733
Baseline: *first*	0.46407	0.48240	0.47294	0.01833
Ledeneva: [7]	0.46576	0.48278	0.47399	0.01702
TextRank: [14]	0.46582	0.48382	0.47450	0.018
Proposed	**0.48155**	**0.47633**	**0.47906**	**0.00522**
TextRank: [15]	0.47207	0.48990	0.48068	0.01783

6 Discussion and Conclusions

In this work, we proposed an extractive automatic text summarization approach by sentence extraction using an unsupervised learning algorithm. In particular, the K-means algorithm for creating groups of similar sentences was used. Then, from the groups of sentences, the most representative sentence was selected for composing the summary. Normally, the definition of the number of groups to form and the initial seeds of the groups are considered as disadvantages of K-means. However, these parameters are used to take advantage of Baseline sentences in order to improve the quality of the summaries. The proposed approach, in contrast to supervised methods, does not need large amount of golden samples for training. Therefore, our proposed approach is more independent from language and dominion.

According to experimental results we demonstrate that the proposed approach obtains more favorable results than others state-of-the-art approaches; ranking our proposed approach in second place, very close to the first place. In addition, our proposed approach outperforms the Baseline (*first*) heuristic for F-measure results, except for 1-gram and BOOL weighting.

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. Song, Y., et al.: A Term Weighting Method based on Lexical Chain for Automatic Summarization. In: Gelbukh, A. (ed.) CICLing 2004. LNCS, vol. 2945, pp. 636–639. Springer, Heidelberg (2004)
3. HaCohen-Kerner, Y., Zuriel, G., Asaf, M.: Automatic Extraction and Learning of Keyphrases from Scientific Articles. In: Gelbukh, A. (ed.) CICLing 2005. LNCS, vol. 3406, pp. 657–669. Springer, Heidelberg (2005)
4. 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)
5. 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)
6. Neto, L., Freitas, A.A., Kaestner, C.A.A.: Automatic Text Summarization using a Machine learning Approach. In: Proceedings of the ACL 2004 Workshop, Barcelona, España (2004)
7. Ledeneva, Y., Gelbukh, A., García, H.R.: Terms Derived from Frequent Sequences for Extractive Text Summarization. In: Gelbukh, A. (ed.) CICLing 2008. LNCS, vol. 4919, pp. 593–604. Springer, Heidelberg (2008)
8. Ledeneva, Y., Gelbukh, A., García, H.R.: Keeping Maximal Frequent Sequences Facilitates Extractive Summarization. Research in Computing Science 34 (2008)
9. Cristea, D., Postolache, O., Pistol, I.: Summarization through Discourse Structure. In: Gelbukh, A. (ed.) CICLing 2005. LNCS, vol. 3406, pp. 632–644. Springer, Heidelberg (2005)
10. Kupiec, J., Pedersen, J.O., Chen, F.: A Trainable Document Summarizer. In: Proc. 18th ACM-SIGIR Conf. on Research and Development in Information Retrieval, pp. 68–73 (1995)

11. DUC. Document Understanding Conference 2002 (2002), http://www-nlpir.nist.gov/projects/duc
12. Xu, W., Li, W., Wu, M., Li, W., Yuan, C.: Deriving Event Relevance from the Ontology Constructed with Formal Concept Analysis. In: Gelbukh, A. (ed.) CICLing 2006. LNCS, vol. 3878, pp. 480–489. Springer, Heidelberg (2006)
13. Mihalcea, R.: Random Walks on Text Structures. In: Gelbukh, A. (ed.) CICLing 2006. LNCS, vol. 3878, pp. 249–262. Springer, Heidelberg (2006)
14. Mihalcea, R., Tarau, P.: TextRank: Bringing Order into Texts. In: Proc. Empirical Methods in Natural Language Processing (EMNLP 2004), Barcelona, Spain (2004)
15. Hassan, S., Mihalcea, R., Banea, C.: Random-Walk Term Weighting for Improved Text Classification. In: Proc. Semantic Computing (ICSC 2007), Irvine, CA (2007)
16. Liu, D., He, Y., Ji, D., Hua, J.: Multi-Document Summarization Based on BE-Vector Clustering. In: Gelbukh, A. (ed.) CICLing 2006. LNCS, vol. 3878, pp. 470–479. Springer, Heidelberg (2006)
17. Bolshakov, I.A.: Getting One's First Million...Collocations. In: Gelbukh, A. (ed.) CICLing 2004. LNCS, vol. 2945, pp. 229–242. Springer, Heidelberg (2004)
18. Koster, C.H.A.: Transducing Text to Multiword Units. In: Workshop on Multiword Units MEMURA at 4th Int. Conf. on Language Resources and Evaluation, LREC 2004, Portugal (2004)
19. Sidorov, G., Gelbukh, A.: Automatic Detection of Semantically Primitive Words Using Their Reachability in an Explanatory Dictionary. In: Proc. Int. Workshop on Natural Language Processing and Knowledge Engineering, NLPKE 2001, USA, pp. 1683–1687 (2001)
20. Luhn, H.P.: A Statical Approach to Mechanical Encoding and Searching of Literary Information. IBM Journal of Research and Development, 309–317 (1975)
21. Salton, G., Buckley, C.: Term-Weighting Approaches in Automatic Text Retrieval. Information Processing & Management 24, 513–523 (1988)
22. Lin, C.Y.: ROUGE: A Package for Automatic Evaluation of Summaries. In: Proceedings of Workshop on Text Summarization of ACL, Spain (2004)
23. Lin, C.Y., Hovy, E.: Automatic Evaluation of Summaries Using N-gram Co-Occurrence Statistics. In: Proceedings of HLT-NAACL, Canada (2003)
24. Spark Jones, K., Willet, P.: Reading in Information Retrieval. Morgan Kaufmann, San Francisco (1997)