AUTOMATIC SUMMARIZATION OF NEWS USING WORDNET CONCEPT GRAPHS

Laura Plaza
Alberto Díaz
Facultad de Informática, Universidad Complutense de Madrid
C/ Prof. José García Santesmases, s/n. 28040 Madrid (Spain)
lplazam@fdi.ucm.es, albertodiaz@fdi.ucm.es

Pablo Gervás
Instituto de Tecnología del Conocimiento, Universidad Complutense de Madrid
C/ Prof. José García Santesmases, s/n. 28040 Madrid (Spain)
pgervas@sip.ucm.es

ABSTRACT

One of the main handicaps in research on automatic summarization is the vague semantic comprehension of the source, which is reflected in the poor quality of the consequent summaries. Using further knowledge, as that provided by ontologies, to construct a complex semantic representation of the text, can considerably alleviate the problem.
In this paper, we introduce an ontology-based extractive method for summarization. It is based on mapping the text to concepts and representing the document and its sentences as graphs. We have applied our approach to news articles, taking advantages of free resources such as WordNet. Preliminary empirical results are presented and pending problems are identified.

KEYWORDS

Automatic Summarization, Graph Theory, Ontology, Natural Language Processing

1. INTRODUCTION

Nowadays, Internet access to news sites has become a day to day practice. But the huge amount of news generated every day makes an exhaustive reading unfeasible. In order to tackle this overload of information, automatic summarization can undoubtedly play a role, allowing users to get a proper idea of what an article is about in just a few lines without having to read the complete item. Some news delivery services already provide summarization tools to support users in selecting relevant information in the news items. Nevertheless, there is much room for improvements.

In past years, a large volume of resources, such as ontologies like WordNet, has emerged. As they intend to provide particular meanings of terms as they apply to the domain in hands, they can definitely benefit the development of NLP systems and, in particular, when used in automatic summarization, they can increase the quality of the resulting summaries.

Automatic document summarization has been an important subject of study since pioneer works by Luhn and Edmundson in the 50s and 60s. While these early approaches were based on simple heuristic features, such as the position of sentences in the document (Brandow *et al*., 1995) or the frequency of the words they contain (Luhn, 1958; Edmundson, 1969), recently, several graph-based methods have been proposed to rank sentences for extraction (Erkan and Radev, 2004).

Several authors (Erkan and Radev, 2004; Mihalcea and Tarau, 2004) have applied the graph theory to text summarization, in order to construct a shallow representation of the documents from text units. However, few approaches explore more complex representations based on concepts connected by semantic relations (synonymy, hypernymy, and similarity relations). One of the main arguments defending the use of shallow

representations is their language independence, while semantic representation provides additional knowledge that can benefit the quality of the resulting summary.

In this paper, we introduce a graph-based approach to extractive summarization for domain-independent documents, which uses ontologies to identify concepts and semantic relations between them and allows a richer text representation. The method proposed uses a semantic graph-based representation for the documents, where vertices are the concepts in WordNet associated to the terms, and the edges indicate different relations between them. This representation makes it possible to combine the desired domain-independence with the use of complex semantic relations.

2. RELATED WORK

Sparck-Jones (Sparck-Jones, 1999) defined a summary as a *reductive transformation of source text to summary text through content reduction by selection and/or generalization on what is important in the source.* This definition may seem obvious, but the truth is that nowadays automatic summarization still exhibits important deficiencies and continues concentrating a considerable body of work.

The definition by Sparck-Jones suggested that there exist two generic groups of summarization methods: those which generate *extracts* and those which generate *abstracts.* Extractive methods construct summaries basically by selecting salient sentences from documents and therefore they are integrally composed of material that is explicitly present in the source. Although human summaries are typically abstracts, most of existing systems produce extracts.

Sentence extractive methods typically build summaries based on a superficial analysis of the source. Early summarization systems were based on simple heuristic features, as the position of sentences in the document (Brandow et al., 1995), the frequency of the words they contain (Luhn, 1958; Edmundson, 1969), or the presence of certain cue words or indicative phrases (Edmundson, 1969). Some advanced approaches also employ machine learning techniques to determine the best set of attributes for extraction (Kupiec et al., 1995).

Recently, several graph-based methods have been proposed to rank sentences for extraction. LexRank (Erkan and Radev, 2004) is an example of a centroid-based method to automatic summarization that assesses sentence importance based on the concept of eigenvector centrality. It assumes a fully connected, undirected graph with sentences as nodes and similarities between them as edges. It represents the sentences in each document by its *tf*idf* vectors and computes sentence connectivity using the cosine similarity. Most recently, Litvak and Last (Litvak and Last, 2008) proposed a novel approach that makes use of a graph-based syntactic representation of text documents for keyword extraction to be used as a first step in summarization.

Even if results are promising, both graph-based approaches exhibit important deficiencies which are consequences of not capturing the semantic relations between terms (synonymy, hypernymy, homonymy, co-occurrence relations, and so on). The following two sentences illustrate these problems.

1. *"Hurricanes are useful to the climate machine. Their primary role is to transport heat from the lower to the upper atmosphere," he said.*
2. *He explained that cyclones are part of the atmospheric circulation mechanism, as they move heat from the superior to the inferior atmosphere.*

As both sentences present different terms, approaches based on term frequencies do not succeed in determining that both have exactly the same meaning. However, methods based on semantic representations indeed capture this equivalence.

3. WORDNET

WordNet is an electronic lexical database developed at Princeton University (Miller et al., 1993). Wordnet[1] structures lexical information in terms of word meanings. Words of the same syntactic category that can be used to express the same concept are grouped into a single synonym set, called synset. Each synset has a

[1] WordNet: http://wordnet.princeton.edu

unique identifier and a gloss that defines the *synset*. Most synsets are connected to other synsets via a number of semantic relations. These relations vary with the type of word, and include among others:

- **Hypernyms and Hyponyms**: *Y* is a hypernym of *X* if every *X* is a *Y* (*feline* is a hypernym of *cat*). *Y* is a hyponym of *X* if every *Y* is a *X* (*cat* is a hyponym of *feline*).
- **Holonym and Meronym**: *Y* is a holonym of *X* if *X* is a part of *Y* (*vehicle* is a holonym of *wheel*). *Y* is a meronym of *X* if *Y* is a part of *X* (*wheel* is a meronym of *vehicle*).
- **Troponym**: the verb *Y* is a troponym of the verb *X* if the activity *Y* is doing *X* in some manner (*to lisp* is a troponym of *to talk*)
- **Entailment**: the verb *Y* is entailed by *X* if by doing *X* you must be doing *Y* (*to sleep* is entailed by *to snore*)

4. SUMMARIZATION METHOD

The method proposed consists of three steps: document representation, concept clustering and sentence selection. Each step is discussed in detail below. A preliminary system has been implemented and tested on several documents from the Document Understanding Conferences 2002[2] collection. In order to clarify how the algorithm works, each step is illustrated in a worked document example from the DUC collection.

4.1 Document representation

In order to construct the concept graph that represents the document, a preliminary step is undertaken to split the text into sentences and to remove generic and high frequency terms. This preprocessing has been carried out using the *Tokenizer, Part of Speech Tagger* and *Sentence Splitter* modules in GATE[3].

Once the sentences have been isolated, we use *WordNet Sense Relate*[4] to translate the terms in each sentence to the appropriate concepts in WordNet. WordNet Sense Relate uses measures of semantic similarity and relatedness to perform word disambiguation. The "*all words*" version assigns a sense or meaning (as found in WordNet) to each word in a text. It carries out Word Sense Disambiguation (WSD) by measuring the semantic similarity between a word and its neighbors (Patwardhan *et al.*, 2005). Figure 1 shows the result of applying WordNet Sense Relate to an example sentence.

Figure 1. WordNet Senses in an example sentence

Hurricane Gilbert swept toward the Dominican Republic Sunday, and the Civil Defense alerted its heavily populated south coast to prepare for high winds, heavy rains and high seas.

Term	WN Sense	Term	WN Sense	Term	WN Sense
hurricane	1	defense	9	prepare	4
Gilbert	2	alert	1	high	2
sweep	1	heavily	2	wind	1
Dominican Republic	1	populate	2	heavy	1
sunday	1	south	1	rain	1
civil	1	coast	1	sea	1

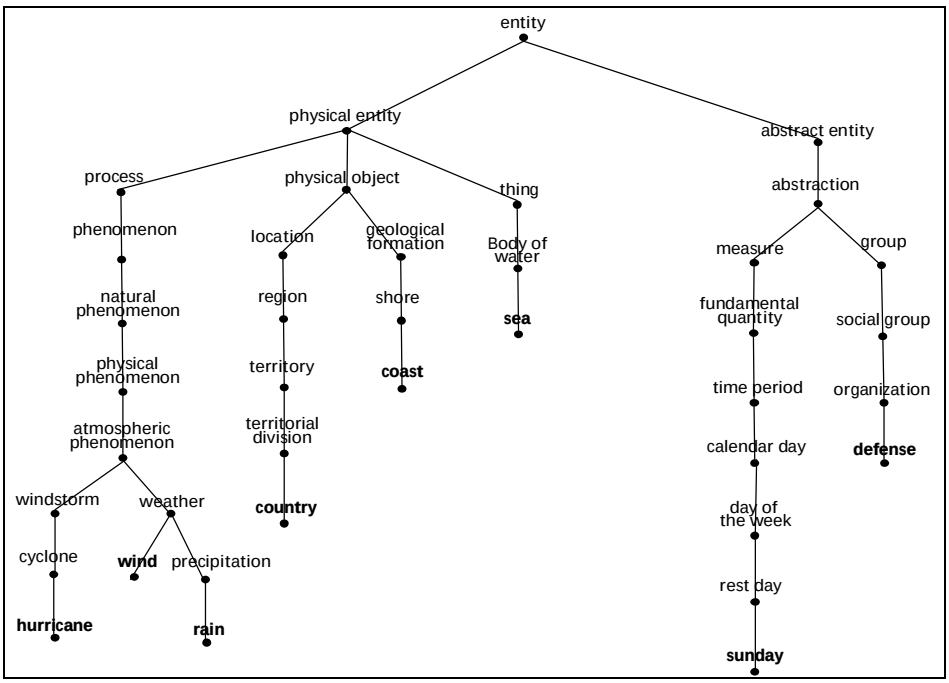
The term *defense* clearly illustrates the importance of using a disambiguation algorithm. The noun *defense* presents 11 different senses in WordNet and, to be precise, the first sense refers to the role of certain players in some sports. Obviously, without a WSD algorithm the wrong sense would be considered.

The resulting WordNet concepts derived from common nouns are extended with their hypernyms, building a hierarchical representation for each sentence in the document, where edges represent semantic relations and they are temporally unlabeled, and only a single vertex is created for each distinct concept in

[2] Document Understanding Conferences (DUC): http://www-nlpir.nist.gov/projects/duc/data.html
[3] GATE (Generic Architecture for Text Engineering): http://gate.ac.uk/
[4] Wordnet Sense Relate: http://www.d.umn.edu/~tpederse/senserelate.html

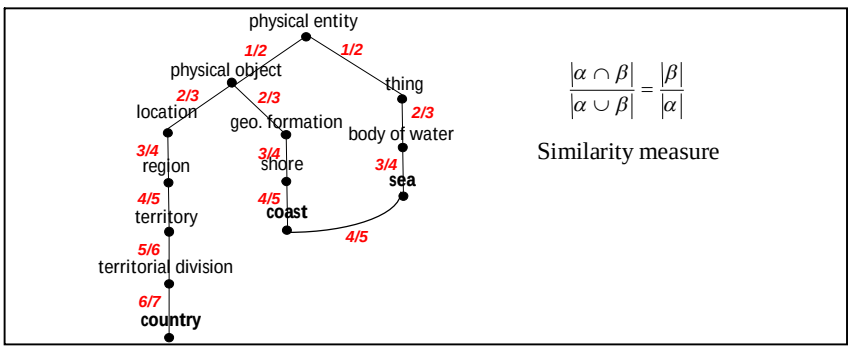
the text. This means that if two different terms in a sentence stand for the same concept, only one vertex is created in the graph that represents both terms. Verbs, adverbs, adjectives and proper nouns are not taken into account in this phase. Figure 2 shows the semantic representation for the example sentence.

Figure 2. Sentence graph



After that, the sentence graphs are merged into a single graph that represents the whole document. This graph can be extended with different and more specific semantic relations between nodes. We have conducted several experiments using a "semantic similarity relation" apart from the *is a* relation previously mentioned. In order to compute this similarity between every pair of leaf concepts in the graph, we use the *WordNet Similarity* package (Pedersen *et al.*, 2004). This package is a Perl module that implements a variety of semantic similarity and relatedness measures based on the information found in the lexical database WordNet. In particular, we have used the *Lesk* algorithm, which computes semantic relatedness of word senses using gloss overlaps (Banerjee and Pedersen, 2002). A new edge is added between two nodes if the underlying concepts are more similar than a *similarity threshold.* Different tests have been conducted in two ways: using together both types of relations (the hypernymy relation and the similarity relation) and using the hypernymy relation on its own, in order to determine the best graph-based representation. It is important to note that it is not feasible to use only the similarity relation, as this will lead to a very disconnected graph and will make the clustering algorithm inadequate.

Figure 3. Example of edge weights assignment



Finally, each edge is labeled with a weight, which is directly proportional to the depth in the hierarchy at which the concepts lies (Figure 3). That is to say, the more specific the concepts connected are, the more weight is assigned to them. Edge weights are calculated using a taxonomy similarity measure (Rada *et al.*, 1989), according to Figure 3, where α is the set of all the parents of a concept, including the concept, and β is the set of all the parents of its immediate higher-level concept, including the concept.

4.2 Concept clustering and subtheme identification

The following step consists of clustering the WordNet concepts in the document graph. The aim is to construct sets of concepts that are closely related in meaning. We presume that each set represents a *subtheme* in the document and that the most central concepts in the cluster give the necessary information related to its subtheme.

We hypothesize that the document graph is an instance of a *scale-free network* (Barabasi and Albert, 1999). These networks present a particular type of nodes which are highly connected to other nodes in the network (*hub nodes)*, while the remaining nodes are quite unconnected. Following (Yoo et al., 2007), we introduce the *salience* of a vertex (v_i) as the sum of the weights of the edges that have as source or target the given vertex (2).

$$salience(v_i) = \sum_{e_j \mid \exists v_k \wedge e_j conecta(v_i, v_k)} weight(e_j) \tag{2}$$

Within the set of vertices, the *n* vertices with a higher salience (*Hub Vertices)* are selected and iteratively grouped in *Hub Vertex Sets (HVS)*. The HVS are set of vertices which are strongly related to one another, and constitute the centroids of the clusters to construct. The remaining vertices (the ones not included in the HVS) are assigned to that cluster to which they are more connected. This is again an iterative process that adjusts the HVS and the vertices assigned progressively.

4.3 Sentence Selection

Once the concept clusters have been created, each sentence is assigned to a cluster. Thus it is necessary to define a similarity measure between a cluster and a sentence graph. As the two representations are quite different in size, traditional graph similarity metrics (i.e. the edit distance) are not convenient and therefore a vote mechanism, adapted from (Yoo *et al.*, 2007) is used (3). Each vertex (v_k) of a sentence (O_j) gives to each cluster (C_i) a different number of votes ($w_{i,j}$) depending on whether the vertex belongs to HVS or non-HVS.

$$similarity(C_i, S_j) = \sum_{v_k \mid v_k \in O_j} w_{k,j}, \quad \text{where} \begin{cases} v_k \notin C_i \Rightarrow w_{k,j} = 0 \\ v_k \in HVS(C_i) \Rightarrow w_{k,j} = 1.0 \\ v_k \notin HVS(C_i) \Rightarrow w_{k,j} = 0.5 \end{cases} \tag{3}$$

Finally, the last step implies the selection of the most significant sentences for the summary, based on the similarity between sentences and clusters as defined in expression (3). The number of sentences to be selected (*N*) varies on the desired compression rate. Three different heuristics have been investigated:

- **Heuristic 1:** For each cluster, the top n_i sentences are selected, where n_i is proportional to its size.
- **Heuristic 2:** We accept the hypothesis that the cluster with more concepts represents the main theme in the document, and select the top *N* sentences from this cluster.
- **Heuristic 3:** We compute a single score for each sentence, as the sum of the votes assigned to each cluster adjusted to their sizes, and select the *N* sentences with higher scores.

5. EXPERIMENTAL RESULTS

Our main objective is to determine the optimal values for the different parameters that get involved in our summarization algorithm. That means we have studied the following research questions:

1. Which of the three heuristics above produces the best summaries?
2. What percentage of vertices should be considered as hub vertices by the clustering method?

3. Is it better to consider both types of relations between concepts (hypernymy and similarity) or just the hypernymy relation?
4. If the similarity relation is taken into account, what similarity threshold should be considered?

In what concerns the evaluation process, as *ROUGE* (Lin, 2004) is the most common metric for automatic evaluation of summarization, we have performed the evaluation by computing the ROUGE-1, ROUGE-2, ROUGE-L and ROUGE-W-1.2 recall metrics. ROUGE (*Recall-Oriented Understudy for Gisting Evaluation*) includes several measures to automatically determine the quality of a summary by comparing it to other (ideal) summaries created by humans. The measures count the number of over- lapping units such as n-gram, word sequences, and word pairs between the computer-generated summary to be evaluated and the ideal summaries created by humans.

In order to answer the questions raised, experiments have been performed on the collection of summarized news items supplied by the Document Understanding Conferences 2002 (DUC, 2002). The collection is composed of 566 news articles in English along with one or more summaries. For the parameter evaluation in hands, only ten news items have been considered. The compression rate has been set to 30%.

The first group of experiments are directed to determine the best of the three heuristics for sentence selection proposed in Section 4.3, along with the percentage of hub vertices for the clustering method, as explained in Section 4.2. For these experiments, only the hypernymy relation has been used. As shown in Table 1, the third heuristic presents slightly better results than the other two, when the percentage of hub vertices is set to 2 percent. Nonetheless, the differences between the three heuristics are not significant. So, a priori, we cannot make a decision on the best heuristic. As long as the number of hub vertices is concerned, the experiments have evidenced that the best setting for this parameter is 2 percent for all heuristics.

Table 1. Heuristic and hub vertices percentage evaluation

		Average R-1	**Average R-2**	**Average R-L**	**Average R-W-1.2**
Heuristic 1	***1- percent***	0,69324	0,33723	0,65202	0,24847
	2- percent	0,71474	0,34520	0,69115	0,26225
	5- percent	0,67814	0,28308	0,63836	0,23646
	10- percent	0,67201	0,27093	0,62717	0,22283
Heuristic 2	***1- percent***	0,71446	0,33185	0,67367	0,25384
	2- percent	0,72487	0,34438	0,68040	0,25810
	5- percent	0,70358	0,30756	0,65924	0,24488
	10- percent	0,72449	0,32887	0,67966	0,25547
Heuristic 3	***1- percent***	0,72056	0,34105	0,68058	0,25727
	2- percent	**0,72755**	**0,34438**	**0,68308**	**0,25886**
	5- percent	0,70560	0,31164	0,66377	0,24612
	10- percent	0,71273	0,31980	0,66888	0,25162

The aim of the second group of tests is to find out if it is better to construct the document graph using just the hypernymy relation between concepts or the hypernymy and similarity relation together (see Section 4.1). For these experiments, the percentage of hub vertices has been set to 2, and the similarity threshold has been temporary established to 0,2. Table 2 manifest that using both relations improves summary evaluation and corroborates that third heuristic is the most effective.

Table 2. Semantic relations evaluation

		Average R-1	**Average R-2**	**Average R-L**	**Average R-W-1.2**
Heuristic 1	***Hypernymy***	0,73474	0,34520	0,69115	0,26225
	Hyp. & Sim.	0,72736	0,34230	0,68558	0,25681
Heuristic 2	***Hypernymy***	0,72487	0,34438	0,68040	0,25810
	Hyp. & Sim.	0,72920	0,33949	0,68463	0,25664
Heuristic 3	***Hypernymy***	0,72755	0,34438	0,68308	0,25886
	Hyp. & Sim.	**0,73118**	**0,32941**	**0,67838**	**0,25323**

Next, the similarity threshold for the WordNet Similarity algorithm must be determined. Table 3 shows the comparisons for different thresholds, when the third heuristic is considered. According to these results, value 0,2 reports the best outcome.

Table 3. Similarity threshold evaluation

	Average R-1	**Average R-2**	**Average R-L**	**Average R-W-1.2**
0.01	0,71145	0,31381	0,66314	0,24872
0.05	0,71470	0,32736	0,67565	0,25250
0.1	0,71953	0,32573	0,67578	0,25477
0.2	**0,73118**	**0,32941**	**0,67838**	**0,25323**
0.5	0,71058	0,31786	0,66690	0,24892

Finally, once the best parametrization has been established, in order to evaluate our method, we calculate a lower bound using a baseline summary constructed by including the first sentences in the document (also known in the literature as the *lead baselineI.* The best summary has been constructed running our method with the best parameter configuration as determined in the experiments above.

Table 4. Comparison with a lead baseline

	Average R-1	**Average R-2**	**Average R-L**	**Average R-W-1.2**
Lead Baseline	0,59436	0,18826	0,55522	0,20488
Best Configuration	0,73118	0,32941	0,67838	0,25323

Table 4 shows that the performance of this method is clearly better than the baseline. It must be taken into account that the positional heuristic used in the baseline seems like a quite pertinent heuristic for the summarization of news articles, where the most important information is usually concentrated in the one or two first sentences. Nonetheless, when dealing with very long documents (as scientific papers) this heuristic is not that appropriate and the difference with sophisticated methods becomes more evident

6. CONCLUSIONS AND FUTURE WORK

In this paper we introduce a method for summarizing text documents. Even if the method is domain-independent, it has been applied to a specific type of documents: news. We represent the document as an ontology-enriched graph, using WordNet concepts and relations. This way we get a richer representation than the one provided by traditional models based on terms which results in a considerable improvement of the evaluation and quality of the resulting summaries.

Another important contribution of the method proposed is the possibility of applying it to documents from different domains. It has been specially designed to work in any domain with minor changes, as it only requires modifying the ontology and the disambiguation algorithm. The authors have previously tested this method in a completely different domain: automatic summarization of biomedical scientific articles (Plaza *et al.*, 2008) with promising results as well.

Nonetheless, we have identified several problems and some possible improvements. First, as our method extracts whole sentences, long ones have higher probability of being selected, because they contain more concepts. The alternative could be to normalize the sentences scores by the number of concepts. Second, in order to formally evaluate the method, a large-scale evaluation is under way on the 566 news articles from the DUC 2002. As future work, we plan to compare these results with those reported by similar systems (i.e. LexRank). This will allow us to determine if significant statistical differences exist between the algorithm presented in this paper and the most accepted methods in the area.

Finally, we are working in an extension of the method to accomplish multi-document summarization, which will permit the generation of a single summary from a set of documents regarding the same topic. Another interesting future work would be to take part in the *update summarization* task, proposed by the Text

Analysis Conference 2009 (TAC, 2009)[5], which involves writing a summary of a set of newswire articles, under the assumption that the user has already read a given set of earlier articles.

ACKNOWLEDGEMENT

This research is funded by the Spanish Ministry of Education and Science (TIN2006-14433-C02-01). This research has been partially funded by UCM and CAM through the IVERNAO project (contract CCG08-UCM/TIC-4300). Also, this research has been partially funded by the Spanish Ministry of Science and Innovation through the FPU program.

REFERENCES

Banerjee, S. & Pedersen, T. 2002. An Adapted Lesk Algorithm for Word Sense Disambiguation using WordNet . In *Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics*, pp. 136-145.

Brandow, R. et al. 1995. Automatic Condensation of Electronic Publications by Sentence Selection. In *Information Processing and Management*. Vol. 5, No. 31, pp. 675–685.

Edmundson, H.P. 1969. New Methods in Automatic Extracting. In *Journal of the Association for Computing Machinery*. Vol. 2, No. 16, pp. 264–285.

Erkan, G. & Radev, D. R. 2004. LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. In *Journal of Artificial Intelligence Research (JAIR)*. No. 22, pp. 457–479.

Lin, C-Y. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In Proceedings *of Workshop on Text Summarization Branches Out, Post-Conference Workshop of ACL*.

Luhn, H.P. 1958. The Automatic Creation of Literature Abstracts. In *IBM Journal of Research Development*. Vol. 2, No. 2, pp. 159–165.

McKeown K.R. et al. 2002. Tracking and summarizing news on a daily basis with Columbia's Newsblaster. In *Proceedings of the second international conference on Human Language Technology Research*, pp. 280-285.

Mihalcea, R. & Tarau, T. 2004. TextRank – bringing order into texts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Barcelona, Spain, pp. 404–411.

Miller, G. A. et al. 1993. Introduction to WordNet: An On-Line Lexical Database. Url: *http://www.cogsci.princeton.edu/~wn/5papers.pdf*, last accessed April , 2009.

Patwardhan, S. et al. 2005. SenseRelate::TargetWord - A Generalized Framework for Word Sense Disambiguation. In *Proceedings of the Twentieth National Conference on Artificial Intelligence (Intelligent Systems Demonstrations)*. Pittsburgh, PA, pp. 1692–1693.

Pedersen, T. et al. 2004. WordNet::Similarity - Measuring the Relatedness of Concepts. In *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics Demonstrations*. Boston, MA, pp. 38–41.

Plaza, L. et al. 2008. Concept-graph based Biomedical Automatic Summarization using Ontologies. In *TextGraphs-3: Graph-based Algorithms for Natural Language Processing. Coling 2008.* Manchester, UK.

Rada, R. et al. 1989. Development and application of a metric on semantic nets. In IEEE Transactions on Systems, Man and Cybernetics, Vol. 19, No. 1, pp. 17-30.

Sparck-Jones, K. 1999. Automatic Summarizing: Factors and Directions. In *Advances in Automatic Text Summarization.* I. Mani and M.T. Maybury Ed. The MIT Press.

Yoo, I. et al. 2007. A coherent graph-based semantic clustering and summarization approach for biomedical literature and a new summarization evaluation method. In *BMC Bioinformatics*. Vol. 8, No. 9.

[5] Text Analysis Conference (TAC): http://www.nist.gov/tac/2009/Summarization/index.html