Procesamiento del Lenguaje Natural, Revista nº 47 septiembre de 2011, pp 97-105 recibido 23-04-2011 aceptado 24-05-2011

Using Semantic Graphs and Word Sense Disambiguation Techniques to Improve Text Summarization

Uso de Grafos Semánticos y de Técnicas de Desambiguación en la Generación Automática de Resúmenes

Laura Plaza
Universidad Complutense de Madrid
Prof. José García Santesmases, s/n
28040 Madrid
lplazam@fdi.ucm.es

Alberto Díaz
Universidad Complutense de Madrid
Prof. José García Santesmases, s/n
28040 Madrid
albertodiaz@fdi.ucm.es

Resumen: En este trabajo se presenta un método para la generación automática de resúmenes basado en grafos semánticos. El sistema utiliza conceptos y relaciones de WordNet para construir un grafo que representa el documento, así como un algoritmo de clustering basado en la conectividad para descubrir los distintos temas tratados en él. La selección de oraciones para el resumen se realiza en función de la presencia en las oraciones de los conceptos más representativos del documento. Los experimentos realizados demuestran que el enfoque propuesto obtiene resultados significativamente mejores que otros sistemas evaluados bajo las mismas condiciones experimentales. Asimismo, el sistema puede ser fácilmente adaptado para trabajar con documentos de diferentes dominios, sin más que modificar la base de conocimiento y el método para identificar conceptos en el texto. Finalmente, este trabajo también estudia el efecto de la ambigüedad léxica en la generación de resúmenes.
Palabras clave: Generación automática de resúmenes, grafos semánticos, desambiguación léxica y semántica, agrupamiento de conceptos

Abstract: This paper presents a semantic graph-based method for extractive summarization. The summarizer uses WordNet concepts and relations to produce a semantic graph that represents the document, and a degree-based clustering algorithm is used to discover different themes or topics within the text. The selection of sentences for the summary is based on the presence in them of the most representative concepts for each topic. The method has proven to be an efficient approach to the identification of salient concepts and topics in free text. In a test on the DUC data for single document summarization, our system achieves significantly better results than previous approaches based on terms and mere syntactic information. Besides, the system can be easily ported to other domains, as it only requires modifying the knowledge base and the method for concept annotation. In addition, we address the problem of word ambiguity in semantic approaches to automatic summarization.
Keywords: Automatic summarization, semantic graphs, word sense disambiguation, concept clustering

1. *Introduction*

The problem of summarizing textual documents has been extensively studied during the past half century. Common approaches include training different machine learning models; computing some simple heuristic rules (such as sentence position or cue words); or counting the frequency of the words in the document to identify central terms. However, these approaches think of words as independent entities that do not interact with other words in their context (the sentence, or even the whole document), which is not the way a human thinks when writing a summary.

Recently, graph-based methods have attracted the attention of the NLP community. These methods have been applied to a wide range of tasks, such as word sense disambiguation (Agirre and Soroa, 2009) or question answering (Celikyilmaz, Thint, and Huang, 2009). Regarding summarization, graph-based methods have typically

ISSN 1135-5948

tried to find salient sentences in the text according to their similarity to other sentences, computing this similarity as the cosine distance between their term vectors (Erkan and Radev, 2004). However, few approaches have dealt with the text at the semantic level, and rarely explore more complex representations based on concepts and semantic relations.

In this paper, we examine the use and strength of concept graphs to identify the central topics covered in a text, as a previous step to rank the sentences for the summary. To this aim, we construct a graph where each sentence is represented by the concepts in WordNet that are found in it, and where the different concepts are interconnected to each other by a number of semantic relations. We identify salient concepts in this graph, based on the detection of hub or core vertices. These concepts constitute the centroids of the clusters that delimitate the different topics in the document. The ranking is based on the presence in the sentences of the most representative concepts for each topic.

Our graph-based method has been evaluated on the Document Understanding Conferences 2002 data[1]. We show that our method performs significantly better than previously published approaches. This work also deals with the problem of word ambiguity, which inevitably arises when trying to map the text to WordNet concepts, and shows that applying a word sense disambiguation algorithm benefit text summarization.

2. Related Work

Text summarization is the process of automatically creating a compacted version of a given text. Content reduction can be addressed by selection and/or by generalization of what is important in the source (Sparck-Jones, 1999). This definition suggests that two generic groups of summarization methods exist: those which generate extracts and those which generate abstracts. In this paper, we focus on extractive methods; that is, those which select sentences from the original document to produce the summary.

Traditional summarization systems typically rank the sentences using simple heuristic features such as the sentence position and the presence of certain cue words or terms that are also found in the headings of the document (Edmundson, 1969; Brandow, Mitze, and Rau, 1995). These attributes are usually weighted and combined using a linear function that assesses a single score for each sentence in the document. Most advanced techniques concern the use of graph-based methods to rank textual units for extraction. This work mainly investigates previous work related to these techniques because the method proposed here clearly falls under this category. Graph-based methods usually represent the documents as graphs, where the nodes correspond to text units (such as words, phrases, sentences or even paragraphs), and the edges represent cohesion relationships between these units, or even similarity measures between them (e.g. the Euclidean distance). Once the graph for the document is created, the salient nodes are located in the graph and used to extract the corresponding units for the summary.

LexRank (Erkan and Radev, 2004) is a well-know example of a centroid-based method to multi-document summarization. It assumes a fully connected and undirected graph, with sentences as nodes and similarity between them as edges. It represents the sentences in each document by its TF-IDF vectors and computes the sentence connectivity using the cosine similarity. A very similar method is proposed by Mihalcea and Tarau (2004) to perform mono-document summarization. As in LexRank, the nodes represent sentences and the edges represent the similarity between them, measured as a function of their content overlap. Most recently, Litvak and Last (2008) proposed an approach that uses a graph-based syntactic representation for keyword extraction, which can be used as a first step in summarization. However, most of these systems ignore the latent semantic associations that exist between the words, both intra and inter-sentence (e.g. synonymy, hypernymy or co-occurrence relations).

Consider the paragraph shown in Figure 1. Approaches based on term frequencies and mere syntactic representations do not succeed in determining that the terms *hurricane* and *cyclone* are synonyms, and that both of them are very close in meaning to the noun phrase *tropical storm*. They do not detect that *Puerto Rico*, *Virgin Islands* and *Dominican Republic* are hyponyms of the broader concept *country*, and that *wind*, *rain* and *high*

[1]DUC Conferences: http://duc.nist.gov/

sea are types of atmospheric conditions usually produced by hurricanes.

> Hurricane Gilbert swept toward the Dominican Republic Sunday, and the Civil Defense alerted its populated south coast to prepare for high winds, heavy rains and high seas. The cyclone was approaching from the southeast with sustained winds of 75 mph gusting to 92 mph. Tropical Storm Gilbert formed in the eastern Caribbean and strengthened into a hurricane Saturday night. The weather service issued a flash flood watch for Puerto Rico and the Virgin Islands until Sunday.

Figure 1: A snippet of a news item that illustrates the need to identify semantic relations between terms

This problem can be partially solved by dealing with concepts instead of terms, and semantic relations instead of lexical or syntactical ones. To this end, some recent works have adapted existing methods to deal with concepts. Reeve, Han, and Brooks (2007) adapt the method of lexical chaining to use biomedical concepts. Zhao, Wu, and Huang (2009) use WordNet concepts and synonyms to represent and expand query words in their graph-based summarizer. Lloret et al. (2008) propose a term frequency based approach combined with textual entailment relations between text snippets, while Steinberger et al. (2007) present a term frequency approach fed with anaphoric information.

All these works have demonstrated that even purely lexical approaches can benefit from different sources of semantic information. Nonetheless, semantic approaches have several shortcomings, mainly due to deficiencies in the knowledge database and problems of word ambiguity. By performing word sense disambiguation (WSD), it is expected that the quality of the summaries will improve. However, to the authors' knowledge, no previous study has investigated the influence of word ambiguity in automatic summarization.

3. Summarization Method

The method presented in this paper consists of 4 main steps: (1) concept identification and sentence representation, (2) document representation, (3) concept clustering and subtheme recognition, and (4) sentence selection. Each step is discussed in detail in the following subsections.

3.1. Concept Identification and Sentence Representation

Before starting with the summarization process, a preliminary step is undertaken in order to prepare the document for the subsequent steps. Irrelevant sections in the document (such as *authors*, *source* or *publication date*) are removed. Generic and high frequency terms are also removed, using a stop list and the inverse document frequency (Sparck-Jones, 1972). The headline/title and body sections in the document are separated. Finally, the text in the body section is split into sentences and the terms are tagged with their part of speech.

Next, each sentence is translated to the appropriate concepts in WordNet, using the WordNet::SenseRelate (WNSR) package[2] (Patwardhan, Banerjee, and Pedersen, 2005). WordNet::SenseRelate uses different measures of semantic similarity and relatedness to perform WSD and assigns a sense or meaning (as found in WordNet) to each word in a text. In particular, in this work the *Lesk* WSD method (Lesk, 1986) is used, which computes semantic relatedness of word senses using gloss overlaps. Table 1 shows the result of applying WNSR to an example sentence. The term defense clearly illustrates the need for 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, while the ninth sense refers to an organization responsible for protecting a country. It is obvious that, without a WSD algorithm, the wrong sense would be considered.

Term	WN Sense	Term	WN Sense
hurricane	1	populate	2
Gilbert	2	south	1
sweep	1	coast	1
Dom. Rep	1	prepare	4
Sunday	1	high	2
civil	1	wind	1
defense	9	heavy	1
alert	1	rain	1
heavily	2	sea	1

Table 1: : WordNet senses found in the 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*

After that, the WordNet concepts derived from nouns are extended with their hypernyms, and the hierarchies of all the concepts

[2]http://www.d.umn.edu/~tpederse/senserelate.html

in the same sentence are merged to build a *sentence graph*. Our experimental results have shown that the use of verbs in this graph decreases the quality of the summaries, while adjectives and adverbs are not included because they do not present the hypernymy relation in WordNet. Finally, the N upper levels of these *is-a* hierarchies are removed, since they represent concepts with a excessively broad meaning. This N value has been empirically set to 3.

3.2. Document Representation

Next, all the sentence graphs are merged into a single *document graph* that represents the whole document. This graph can be extended with more specific semantic relations in order to obtain a more complete representation of the document. We have conducted several experiments using a *semantic similarity* relation apart from the *is-a* relation previously mentioned. To this end, we compute the similarity between every pair of leaf concepts in the graph, using the WordNet Similarity package[3] (Banerjee and Pedersen, 2002). This package implements a variety of semantic similarity and relatedness measures based on the information found in WordNet. In particular, we have used the Lesk measure. To expand the document graph with these additional relations, a new edge is added between two leaf nodes if the similarity between the underlying concepts exceeds a *similarity threshold*.

Finally, each edge is assigned a weight in [0, 1]. This weight is calculated as the ratio between the relative positions in their corresponding hierarchies of the concepts linked by the edge (that is, the more specific the concepts connected are, the more weight is assigned to it).

Figure 2 shows an example of an extended document graph for a fictitious document that consists solely of the sentence presented in Table 1. Continuous lines represent is-a relations, while dashed lines represent semantic similarity relations. The edges of a portion of this graph have been labeled with their weights. Ignored too general concepts are shown in a lighter color.

[3]http://wn-similarity.sourceforge.net/

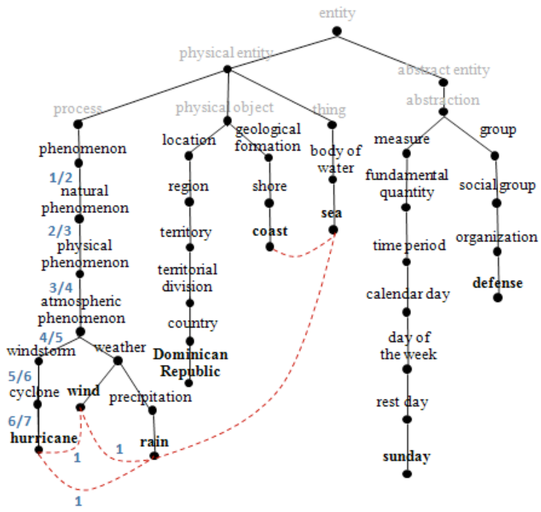

Figure 2: Example of a simplified document graph

3.3. Concept Clustering and Sub-theme Identification

The following step consists in clustering the WordNet concepts in the document graph, using a degree-based clustering algorithm similar to that proposed in (Yoo, Hu, and Song, 2007). The aim is to construct sets of concepts that are closely related in meaning, under the assumption that each set represents a different *sub-theme* in the document and that the most central concepts in the cluster (the centroids) give the necessary and sufficient information related to its subtheme.

We hypothesize that the document graph is an instance of a scale-free network (Barabási and Albert, 1999). A scale-free network is a complex network that (among other characteristics) presents a particular type of nodes which are highly connected to other nodes in the network, while the remaining nodes are quite unconnected. These highest degree nodes are often called hubs.

Following (Yoo, Hu, and Song, 2007), we introduce the salience of a vertex (v_i) as the sum of the weights of the edges connected to v_i (equation 1).

$$salience(v_i) = \sum_{\substack{\forall e_j | \exists v_k \\ \wedge e_j connect(v_i, v_k)}} weight(e_j) \qquad (1)$$

The vertices with highest salience are named *hub vertices*, and they represent the central nodes in the graph. The clustering algorithm starts sorting the vertices by their salience, and selecting the first n vertices in the ranking (that is, the so called hub vertices). Next, the hub vertices are iteratively grouped forming *hub vertex sets*. A hub vertex set (HVS) is a set of vertices strongly connected to one another. These will constitute the centroids of the clusters. To construct these HVSs, the clustering algorithm first searches, iteratively and for each hub vertex, the hub vertex most connected to it, and merges them into a single HVS. In a second stage, the algorithm checks, for every pair of HVSs, if their internal connectivity is lower than the connectivity between them. If so, both HVSs are merged. This decision is encouraged by the assumption that the clustering should show maximum intra-cluster connectivity but minimum inter-cluster connectivity.

Finally, the remaining vertices (those not included in the HVSs) are assigned to that cluster to which they are more connected, as shown in equation 2. This is again an iterative process that adjusts the HVSs and the vertices assigned to them.

$$conn(v, HVS_i) = \sum_{\substack{\forall e_j | \exists w \in HVS_i \\ \wedge e_j\, connect(v,w)}} weight(e_j) \quad (2)$$

3.4. Sentences Selection

Once the concept clusters have been created, we compute the similarity between all sentences in the document and each of these clusters. The similarity between a sentence graph and a cluster is calculated using a non-democratic vote mechanism, so that each vertex (v_k) of a sentence (S_j) gives to each cluster (C_i) a different number of votes ($w_{i,j}$) depending on whether v_k belongs or not to the HVS of that cluster. The similarity is computed as the sum of the votes given by all vertices in the sentence to each cluster, as expressed in equation 3. Next, each sentence is assigned to the cluster to which this similarity is greater.

$$semantic\_similarity(C_i, S_j) = \sum_{v_k | v_k \in S_j} w_{k,j} \quad (3)$$

$$\text{where} \begin{cases} w_{k,j}=0\; si\; v_k \notin C_i \\ w_{k,j}=\gamma\; si\; v_k \in HVS(C_i) \\ w_{k,j}=\delta\; si\; v_k \notin HVS(C_i) \end{cases}$$

Finally, the most significant sentences are selected for the summary, based on the similarity between them and the clusters as defined in equation 3. Three different heuristics for sentence selection have been investigated.

- **Heuristic 1:** Under the hypothesis that the cluster with more concepts represents the main theme in the document, and hence the only one that contributes to the summary, the N sentences with greater similarity to this cluster are selected.
- **Heuristic 2:** All clusters contribute to the summary proportionally to their sizes. Therefore, for each cluster, the top n_i sentences are selected, where n_i is proportional to its size. So, this heuristic will generate summaries covering not only the information related to the main topic, but also other *satellite* information.
- **Heuristic 3:** Halfway between the two heuristics above, this one computes a single score for each sentence as the sum of their similarity to each cluster adjusted to their sizes (equation 4). Then, the N sentences with higher scores are selected.

$$sem\_sim(S_j) = \sum_{C_i} \frac{similarity(C_i, S_j)}{|C_i|} \quad (4)$$

Note that the N value varies with the desired compression rate.

Two additional features, apart from the semantic-graph similarity (*Sem_Graphs*), have been extracted and tested when computing the score of the sentences: sentence location (*Loc*) and similarity with the title section (*Tit*). Despite of their simplicity, these features are commonly used in the most recent works on extractive summarization (Bossard, Généreux, and Poibeau, 2008; Bawakid and Oussalah, 2008). The final selection of sentences is based on the weighted sum of these features, as stated in equation 5.

$$Score(S_j) = \lambda \times Sem\_Graphs(S_j) + \theta \times Loc(S_j) + \chi \times Tit(S_j) \quad (5)$$

4. Evaluation Framework

4.1. Evaluation Metrics: ROUGE

We follow the ROUGE metrics and the guidelines observed in the 2004 and 2005 Document Understanding Conferences (Litkowski, 2004). ROUGE (Lin, 2004) compares a summary generated from an automated system (called *peer*) with one or more ideal summaries (called *models*), usually created by humans, and computes a set of different measures to automatically determine the content quality of the summary. In this work, the ROUGE-1, ROUGE-2, ROUGE-L and ROUGE-S4 recall scores are used to evaluate the summarizer. In short, ROUGE-N evaluates n-grams occurrence, where N stands for the length of the n-gram. ROUGE-L computes the union of the longest common subsequences (LCS) between the candidate and the model summary sentences. Finally, ROUGE-S4 evaluates "skip bigrams", that is, pairs of words having intervening word gaps no larger than four words.

4.2. Evaluation Collection

We adopt the evaluation corpus of DUC 2002, which is the most recent one for single document summarization. This collection is composed of 567 news articles in English. Each document comes with one or more abstractive model summaries manually created by humans. Model summaries are approximately 100 words long. Since the news items have been selected from different sections of different newspapers, the topics covered in the collection are diverse.

4.3. Algorithm Parametrization

Before the final evaluation, a preliminary experimentation has been performed to determine the best configuration for the summarization algorithm. To this end, we use a set of 10 documents from the DUC corpus. The model summaries for these documents were manually created by selecting the 30 % of the most salient sentences in them. So, the model summaries for the parametrization are extractive summaries. The parameters to be estimated include:

- The percentage of vertices considered as hub vertices in the clustering method (see Section 3.3).
- The set of semantic relations used to build the graph (see Section 3.2).
- If the semantic similarity relation is finally used, the similarity threshold to be considered (see Section 3.2).
- The combination of summarization features used to select the sentences and their weights (see Section 3.4).

As a result, we get the optimal parametrization for each of the three heuristics for sentence selection implemented in our system, as shown in Table 2 (Plaza, Diaz, and Gervas, 2010).

Parameter	**H.1**	**H.2**	**H.3**
Percentage of hubs	2 %	20 %	5 %
Set of relations	hypernymy + sem. sim.		
Similarity threshold	0.01	0.05	0.01
Summarization criteria	Sem_Graphs + Location		

Table 2: Summary of the evaluation accomplished to determine the optimal parametrization for the algorithm

It may be seen that the best configuration implies using both relations (*is-a* and semantic similarity), but the percentage of hub vertices and the similarity threshold depend on the heuristic. Heuristics 1 and 3 prefer a relatively small number of hub vertices (2 % and 5 %, respectively), while heuristic 2 prefers a higher number of hub vertices (20 %). This is due to the nature of the summaries generated by the second heuristic. It is worth remembering that the aim of Heuristic 2 is to generate summaries covering all topics presented in the source document, regardless of their relative relevance within the document. Thus, it is not sufficient to consider only the concepts dealing with the main document topic as hub vertices, but also those dealing with other secondary information. The similarity threshold is also higher for this second heuristic than for the remaining ones. On the other hand, the use of the positional criteria, together with our semantic graph-based approach, improves the results obtained by all heuristics and achieves better ROUGE scores than any other combination of sentence selection criteria. This result was expected since the information in news items is usually presented according to the *inverted pyramid* form, so that the most important information is placed first. In particular, the best results are achieved when the parameters λ, θ and χ in equation 5 are set to 0.9, 0.1 and 0.0 respectively.

The choice of the parameters also influences the structural characteristic of the document graph as well as the result of the clustering algorithm. Table 3 shows how the number and size of the clusters are affected by the percentage of hub vertices and the similarity threshold. It can be observed that raising the number of hub vertices increases the number of clusters, but decreases their average size. On the contrary, increasing the connectivity of the graph (i.e. reducing the similarity threshold) decreases the number of clusters, but its effect on the cluster size is unclear.

Sim. Thres.	Hubs	Clusters (HVS)	Larger cluster	Smaller cluster
0.001	2 %	1,33	254,89	9,94
	10 %	6,56	135,78	7,39
	20 %	11,75	77,56	3,75
0.01	2 %	1,79	288,21	13,32
	10 %	6,58	136,11	8,74
	20 %	13,16	76,37	3,58
0.5	2 %	2,37	191,84	16,68
	10 %	7,63	91,37	5,95
	20 %	14,63	54,52	2,37

Table 3: Average number and size of the clusters built from the document graph, according to the similarity threshold and the percentage of hub vertices used

5. *Results and Discussion*

Table 4 shows the ROUGE scores for the summaries created by the three versions of our system (H.1, H.2, H.3); the LexRank[4] lexical graph-based summarizer (Erkan and Radev, 2004); a lexical summarizer improved with anaphoric information (LeLSA+AR) (Steinberger et al., 2007); a term frequency summarizer improved with textual entailment (TextEnt) (Lloret et al., 2008); and the 5 systems which participated in DUC-2002 and achieved the best results (in terms of the ROUGE metric). In short, system 19 uses topic representation templates to extract salient information; systems 21, 27 and 28 employ machine learning techniques to determine the best set of attributes for extraction (word frequency, sentence position...); and system 29 uses lexical chains. We also list a lead baseline (the first 100 words of a document). All summaries were truncated to 100 words as traditionally done in DUC. The highest result for each metric is shown in bold.

System	R-1	R-2	R-S4	R-L
H.3	0,4648	**0,2196**	**0,1928**	**0,4277**
H.2	**0,4651**	0,2193	0,1927	0,4276
H.1	0,4641	0,2191	0,1919	0,4268
LexRank	0,4558	0,2115	0,1846	0,4173
TextEnt	0,4518	0,1942	—	0,4104
LeLSA+AR	0,4228	0,2074	0,1661	0,3928
DUC 28	0,4278	0,2177	0,1732	0,3865
DUC 21	0,4149	0,2104	0,1655	0,3754
Lead	0,4113	0,2108	0,1660	0,3754
DUC 19	0,4082	0,2088	0,1638	0,3735
DUC 27	0,4052	0,2022	0,1600	0,3691
DUC 29	0,3993	0,2006	0,1576	0,3617

Table 4: ROUGE scores for the different versions of our algorithm, and comparison with related work. The systems are sorted by ROUGE-L score in descending order

A Wilcoxon signed ranks test has shown that, at the 95 % confidence level, the performance of our three heuristics is significantly better than that of LexRank, all the DUC systems and both baselines (in at least 2 out of the 4 ROUGE scores). But no significant differences exist between the three heuristics. Regarding the anaphoric and textual entailment approaches, as we only know their average ROUGE scores, we could not apply the test for these systems. However, the three versions of our summarizer outperform the LeLSA+AR system in all ROUGE scores, and the Freq+TextEnt system in ROUGE-1, ROUGE-2 and ROUGE-L scores (the ROUGE- S4 score is not available).

A further experiment has been conducted to examine the effect of WSD on the results reported by our method. To this aim, we repeated these experiments without performing WSD, but simply assigning to each word its first sense in WordNet. The results are presented in Table 5, and indicate that the use of word disambiguation improves the quality of the automatic summaries. The WSD algorithms identify the concepts that are being referred to in the documents more accurately which leads to the creation of a graph that better reflects the content of the document. However, this improvement is less than expected. The reason seems to be that the first WordNet sense criterion is a quite pertinent one, since the senses of the words in WordNet are ranked according to their frequency.

[4]We use the implementation of LexRank as provided in the MEAD summarization platform (http://www. summarization.com/ mead/). The parameters are set to their default values.

Besides, the Lesk algorithm introduces some noise, and it is biased toward the first sense, so that the percentage of WordNet concepts in the DUC corpus that Lesk labels with the first sense is above 61 %. Therefore, the difference among the disambiguation performed by both criteria is not marked.

System	R-1	R-2	R-S4	R-L
H.3-WSD	0,4648	**0,2196**	**0,1928**	**0,4277**
H.2-WSD	**0,4651**	0,2193	0,1927	0,4276
H.1-WSD	0,4641	0,2191	0,1919	0,4268
H.3-1st sense	0,4608	0,2103	0,1838	0,4251
H.2-1st sense	0,4594	0,2073	0,1810	0,4224
H.1-1st sense	0,4584	0,2057	0,1794	0,4216

Table 5: ROUGE scores achieved by our system: first, using Lesk to solve word ambiguity; and second, selecting the 1st sense in WordNet for every word

It is striking that the differences between the three heuristics (both with and without WSD) are not significant. In order to understand the reason, we examined the intermediate results of our algorithm. We found that, in this particular experimentation, the clustering method usually produces one big cluster along with a variable number of small clusters. As news items have little redundancy in their content, most concepts in them are closely related to the main topic, and so they fall into the same cluster. As a consequence, the three heuristics extract most of their sentences from this large cluster, and therefore the summaries are quite similar. However, the best results are reported by the heuristic 3. We have checked that this heuristic selects most of the sentences from the most populated cluster, but it also includes some sentences from others when the sentences give a high score to these clusters. Thus, in addition to the information related to the central topic, this heuristic also includes other dependent or “satellite” information that might be relevant to the user. On the contrary, heuristic 1 fails to present this information; while heuristic 2 includes more secondary information, but misses other core one.

6. Conclusion and Future Work

In this paper, an efficient approach to extractive text summarization has been presented, which represents the document as a semantic graph, using WordNet concepts and relations. The method succeeds in identifying the salient concepts in the text and the central topics covered in it; thus the selection of sentences is close to that made by humans.

An extensive evaluation has been accomplished, which has confirmed that the use of concepts rather than terms, along with the semantic relations that exist between them, can be very useful in automatic summarization. As a result, the method proposed compares positively with previous approaches based on terms. Our results also outperform those obtained by a lexical graph-based approach and by others systems using different types of syntactical information.

We found that applying WSD improves the performance of our summarizer but, as already mentioned, the disambiguation algorithm introduces some noise in the concept recognition, which in turns affects the subtheme identification step. As future work, we plan to evaluate our system with other disambiguation algorithms.

Finally, an important contribution is the possibility of applying the method to documents from different domains with minor changes, as it only requires modifying the knowledge base and the method for automatically identifying the concepts within the text.

References

Agirre, E. and A. Soroa. 2009. Personalizing PageRank for Word Sense Disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 33–41.

Banerjee, S. and T. Pedersen. 2002. An Adapted Lesk Algorithm for Word Sense Disambiguation using WordNet. In *Proceedings of the 3rd International Conference on Intelligent Text Processing and Computational Linguistics*, pages 136–145.

Barabási, A.L. and R. Albert. 1999. Emergence of Scaling in Random Networks. *Science*, 268:509–512.

Bawakid, A. and M. Oussalah. 2008. A Semantic Summarization System: University of Birmingham at TAC 2008. In *Proceedings of the First Text Analysis Conference.*

Bossard, A., M. Génereux, and T. Poibeau. 2008. Description of the LIPN Systems at

TAC 2008: Summarizing Information and Opinions. In *Proceedings of the 1st Text Analysis Conference*.

Brandow, R., K. Mitze, and L. F. Rau. 1995. Automatic Condensation of Electronic Publications by Sentence Selection. *Information Processing and Management*, 5(31):675–685.

Celikyilmaz, Asli, Marcus Thint, and Zhiheng Huang. 2009. A Graph-based Semi-Supervised Learning for Question-Answering. In *Proceedings of the 47th Annual Meeting of the ACL*, pages 719–727.

Edmundson, H. P. 1969. New Methods in Automatic Extracting. *Journal of the Association for Computing Machinery*, 2(16):264–285.

Erkan, G. and D. R. Radev. 2004. LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. *Journal of Artificial Intelligence Research*, 22:457–479.

Lesk, M. 1986. Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from a Ice Cream Cone. In *Proceedings of Special Interest Group on Design of Communication*, pages 24–26.

Lin, C-Y. 2004. Rouge: A Package for Automatic Evaluation of Summaries. In *Proceedings of the Association for Computational Linguistics, Workshop: Text Summarization Branches Out*, pages 74–81.

Litvak, M. and M. Last. 2008. Graph-based Keyword Extraction for Single-document Summarization. In *Proceedings of the International Conference on Computational Linguistics, Workshop on Multi-source Multilingual Information Extraction and Summarization*.

Lloret, E., O. Ferrández, R. Muñoz, and M. Palomar. 2008. A Text Summarization Approach under the Influence of Textual Entailment. In *Proceedings of the 5th International Workshop on Natural Language Processing and Cognitive Science in Conjunction with the 10th International Conference on Enterprise Information Systems*, pages 22–31.

Mihalcea, R. and P. Tarau. 2004. TextRank: Bringing Order into Texts. In *Proceedings of the Conference on Empirical Methods on Natural Language Processing*, pages 404–411.

Patwardhan, S., S. Banerjee, and T. Pedersen. 2005. SenseRelate::TargetWord: A Generalized Framework for Word Sense Disambiguation. In *Proceedings of the Association for Computational Linguistics*, pages 73–76.

Plaza, L., A. Diaz, and P. Gervas. 2010. Automatic Summarization of News Using WordNet Concept Graphs. *IADIS International Journal on Computer Science and Information Systems*, V:45–57.

Reeve, L. H., H. Han, and A. D. Brooks. 2007. The Use of Domain-specific Concepts in Biomedical Text Summarization. *Information Processing and Management*, 43:1765–1776.

Sparck-Jones, K. 1972. A Statistical Interpretation of Term Specificity and its Application in Retrieval. *Journal of Documentation*, 28(1):11–20.

Sparck-Jones, K. 1999. *Automatic Summarising: Factors and Directions*. The MIT Press.

Steinberger, J., M. Poesio, M. A. Kabadjov, and K. Jezek. 2007. Two Uses of Anaphora Resolution in Summarization. *Information Processing and Management*, 43(6):1663–1180.

Yoo, I., X. Hu, and I-Y. Song. 2007. A Coherent Graph-based Semantic Clustering and Summarization Approach for Biomedical Literature and a New Summarization Evaluation Method. *BMC Bioinformatics*, 8(9).

Zhao, L., L. Wu, and X. Huang. 2009. Using Query Expansion in Graph-based Approach for Query-focused Multi-document Summarization. *Information Processing and Management*, 45:35–41.