Semantic Graphs Derived from Triplets with Application in Document Summarization

Delia Rusu
Technical University of Cluj-Napoca, Faculty of Automation and Computer Science,
G. Bariţiu 26-28, 400027 Cluj-Napoca, Romania
E-mail: delia.rusu@gmail.com

Blaž Fortuna, Marko Grobelnik and Dunja Mladenić
Jozef Stefan Institute,
Jamova 39, 1000 Ljubljana, Slovenia
E-mail: {blaz.fortuna,marko.grobelnik, dunja.mladenic}@ijs.si

Keywords: natural language processing, text mining, semantic graph, document summarization

Received: November 12, 2008

Information nowadays has become more and more accessible, so much as to give birth to an information overload issue. Yet important decisions have to be made, depending on the available information. As it is impossible to read all the relevant content that helps one stay informed, a possible solution would be condensing data and obtaining the kernel of a text by automatically summarizing it. We present an approach to analyzing text and retrieving valuable information in the form of a semantic graph based on subject-verb-object triplets extracted from sentences. Once triplets have been generated, we apply several techniques in order to obtain the semantic graph of the document: co-reference and anaphora resolution of named entities and semantic normalization of triplets. Finally, we describe the automatic document summarization process starting from the semantic representation of the text. The experimental evaluation carried out step by step on several Reuters newswire articles shows a comparable performance of the proposed approach with other existing methodologies. For the assessment of the document summaries we utilize an automatic summarization evaluation package, so as to show a ranking of various summarizers.

Povzetek: V članku predlagamo pristop k analizi besedila in zajemanju pomembnih informacij v obliki semantičnega grafa, ki je zasnovan na predstavitvi stavka s trojkami (osebek-povedek-predmet).

1 Introduction

The accessibility of information arises mostly from the rapid development of the World Wide Web and online information services. One has to read a considerable amount of relevant content in order to stay updated, but it is impossible to read everything related to a certain topic. A feasible solution to this admitted problem is condensing this vast amount of data and extracting only the essence of the message, in the form of an *automatically generated summary*.

In this paper we describe a method of text analysis with the stated purpose of extracting valuable information from documents. We shall attach a graphical representation, called semantic graph, to the initial document. The graph is based on triplets retrieved from the document sentences. Moreover, we are going to describe an application of semantic graphs generation– text summarization – as a method for reducing the quantity of information but preserving one important characteristic –its quality.

The paper is organized as follows. Firstly, the triplet based semantic graphs generation algorithm is presented. Two steps are detailed in this phase: triplet extraction from sentences, followed by the procedure of yielding the semantic graph of the document. In order to obtain the graph, named entity co-reference and anaphora resolution as well the semantic normalization of triplets are employed. Secondly, the summarization process is explained, followed by an evaluation of the system components. The paper concludes with several remarks.

2 Triplet based semantic graphs

In English, the declarative sentence has the basic form *subject – verb – object*. Starting from this observation, one can think of the "core" of a sentence as a *triplet* (consisting of the aforementioned three elements). We assume that it contains enough information to describe the *message of a sentence*. The usefulness of triplets resides in the fact that it is much easier to process them instead of dealing with very complex sentences as a whole.

For triplet extraction, we apply the algorithm for obtaining triplets from a treebank parser output described in [1] , and employ the *Stanford Parser* [2] .

The extraction is performed based on pure syntactic analysis of sentences. For obtaining semantic information, we first annotate the document with *named entities*. Throughout this paper, the term *"named entities"* refers to names of people, locations and organizations. For named entity extraction we consider *GATE* (*General Architecture for Text Engineering*) [3] , which was used as a toolkit for natural language processing.

The *semantic graph* corresponds to a visual representation of a document's semantic structure. The starting point for deriving semantic graphs was [4] .

The procedure of semantic graph generation consists of a series of sequential operations composing a pipeline:

- *Co-reference* resolution by employing text analysis and matching methods, thus consolidating named entities.
- *Pronominal anaphora* resolution based on named entities.
- *Semantic normalization* using WordNet synsets.
- *Semantic graph generation* by merging triplet elements with respect to the synset they belong to.

The following sub-sections will further detail these pipeline components.

2.1 Co-reference Resolution

Co-reference is defined as the identification of surface terms (words within the document) that refer to the same entity [4] . For simplification, we are going to consider co-reference resolution for the named entities only. The set of operations we have to perform is threefold. Firstly we have to determine the named entity *gender*, so as to reduce the search space for candidates. Secondly, in the case of named entities composed of more than one word, we eliminate the set of English stop words (for example Ms., Inc., and so on). Thirdly, we apply the heuristics proposed in [4] : two different surface forms represent the same named entity if one surface form is completely included in the other. For example, *"Clarence"*, *"Clarence Thomas"* and *"Mr. Thomas"* refer to the same named entity, that is, *"Clarence Thomas"*. Moreover, abbreviations are also co-referenced, for example *"U.S."*, *"U.S.A."*, *"United States"* and *"United States of America"* all refer to the same named entity – *"United States America"* (*"of"* will be eliminated, as it is a stop word).

2.2 Anaphora Resolution

In linguistics, *anaphora* defines an instance of an expression that refers to another expression; pronouns are often regarded as anaphors. The pronoun subset we considered for anaphora resolution is formed of: {*I, he, she, it, they*}, and their objective, reflexive and possessive forms, as well as the relative pronoun *who*.

We perform a sequential search, first backward and then forward, with the purpose of finding good replacement candidates for a given pronoun, among the named entities. Firstly, we search backwards inside the sentence where we found the pronoun. We select candidates that agree in gender with the pronominal anaphor, as suggested in [5, 6]. Next, we look for possible candidates in the sentences preceding the one where the pronoun is located. If we have found no candidates so far, we search forward within the pronoun sentence, and then forward in the next sentences, as in [4] . Once the candidates have been selected, we apply antecedent indicators to each of them, and assign scores (0, 1, and 2). The antecedent indicators we have taken into account are a subset of the ones mentioned in [5] : *givenness, lexical reiteration, referential distance, indicating verbs* and *collocation pattern preference.* After assigning scores to the candidates found, we select the candidate with the highest overall score as the best replacement for the pronoun. If two candidates have the same overall score, we prefer the one with a higher collocation pattern score. If we cannot make a decision based on this score, we choose the candidate with a greater indicating verbs score. In case of a tie, we select the most recent candidate (the one closest to the pronoun).

We summarize the anaphora resolution procedure in the algorithm in Figure 2.1.

```
function ANAPHORA-RESOLUTION (pronoun, number_of_sentences) returns a solution, or failure
    candidates ← BACKWARD-SEARCH-INSIDE-SENTENCE (pronoun) ∪ BACKWARD-SEARCH (pronoun, number_of_sentences)
    if candidates ≠ ∅ then
    APPLY-ANTECEDENT-INDICATORS (candidates)
    else
    candidates ← FORWARD-SEARCH-INSIDE-SENTENCE (pronoun) ∪ FORWARD-SEARCH (pronoun, number_of_sentences)
    if candidates ≠ ∅ then
    APPLY-ANTECEDENT-INDICATORS (candidates)
    result ← MAX-SCORE-CANDIDATE (candidates)
    if result ≠failure then return result
    else return failure

function APPLY-ANTECEDENT-INDICATORS (candidates) returns a solution, or failure
    result ← APPLY-GIVENNESS (candidates) ∪
    APPLY-LEXICAL-REITERATION (candidates) ∪
    APPLY-REFERENTIAL-DISTANCE (candidates) ∪
    APPLY-INDICATING-VERBS (candidates) ∪
    APPLY-COLLOCATION-PATTERN-PREFERENCE (candidates)
    if result ≠failure then return result
    else return failure
```

Figure 2.1: The anaphora resolution algorithm.

2.3 Semantic Normalization

Once co-reference and anaphora resolution have been performed, the next step is semantic normalization. We compact the triplets obtained so far, in order to generate a more coherent semantic graphical representation. For this task, we rely on the synonymy relationships between words. More precisely, we attach to each triplet element the synsets found with WordNet. If the triplet element is composed of two or more words, then for each of these words we determine the corresponding synsets. This

procedure will help in the next phase, when we merge the triplet elements that belong to the same synset.

2.4 Semantic Graph Generation

Based on the semantic normalization procedure, we can merge the subject and object elements that belong to the same normalized semantic class. Therefore, we generate a directed semantic graph, having as nodes the subject and the object elements and as edges the verbs. Verbs label the relationship between the subject and the object nodes in the graph. An example of a semantic graph obtained from a news article is shown in Figure 3.1.

3 Document summarization

The purpose of summarization based on semantic graphs is to obtain the most important sentences from the original document by first generating the document semantic graph and then using the document and graph features to obtain the document summary [4] . We created a semantic representation of the document, based on the logical form triplets we have retrieved from the text. For each generated triplet we assign a set of features comprising linguistic, document and graph attributes. We then train the linear SVM classifier to determine those triplets that are useful for extracting sentences which will later compose the summary. As features for learning, we select the logical form triplets characterized by three types of attributes: *linguistic attributes*, *document attributes*, *graph attributes*. The *linguistic attributes* include, among others, the triplet type – subject, verb or object – the treebank tag and the depth of the linguistic node extracted from the treebank parse tree and the part of speech tag. The *document attributes* include the location of the sentence within the document, the triplet location within the sentence, the frequency of the triplet element, the number of named entities in the sentence, the similarity of the sentence with the centroid (the central words of the document), and so on. Finally, the *graph attributes* consist of hub and authority weights [7] , page rank [8] , node degree, the size of the weakly connected component the triplet element belongs to, and others.

Features are ranked, based on information gain, and the order is as follows (starting with the most important

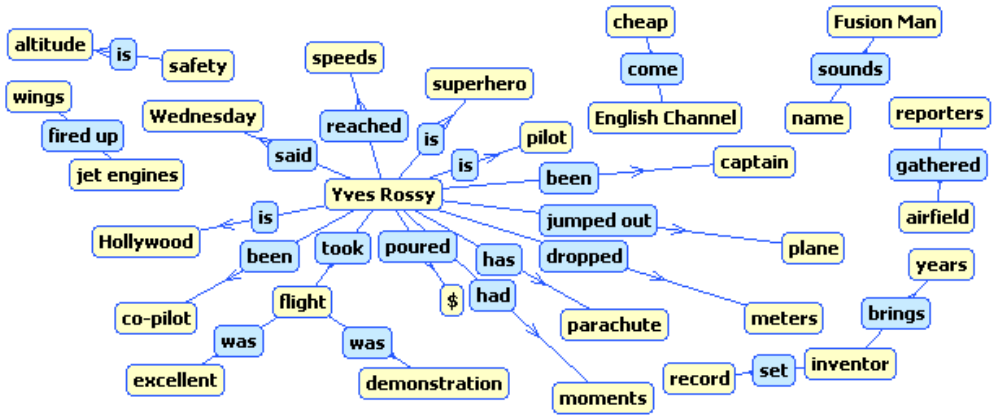

Figure 3.1: A semantic graph obtained from a news article.

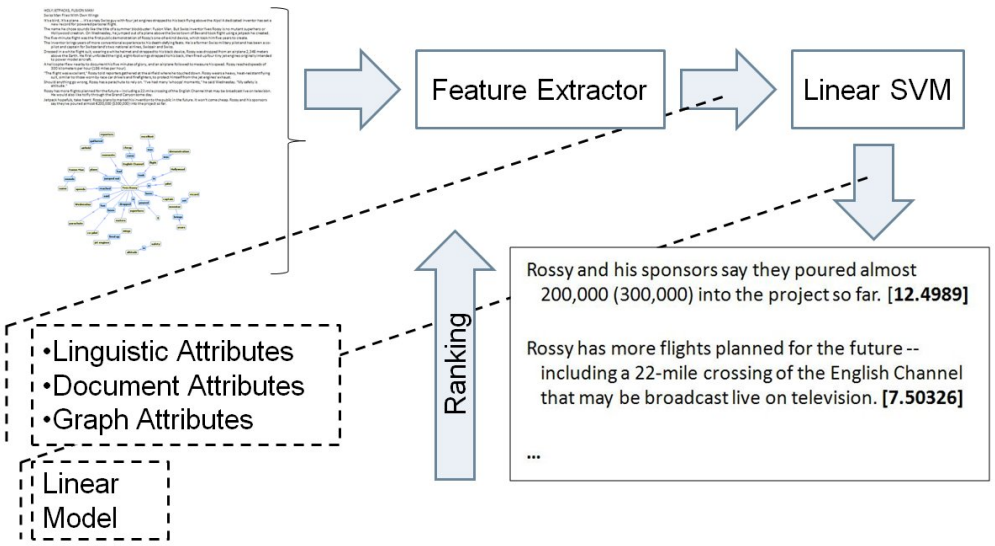

Figure 3.1: The summarization process.

feature): object, subject, verb (all of these are words), location of the sentence in the document, similarity with the centroid, number of locations in the sentence, number of named entities in the sentence, authority weight for the object, hub weight for the subject, size of the weakly connected component for the object.

The summarization process, described in Figure 3.1, starts with the original document and its semantic graph. The three types of features abovementioned are then retrieved. Further, the sentences are classified with the linear SVM and the document summary is obtained. Its sentences are labelled with SVM scores and ordered based on these scores in a decreasing manner. The motivation for doing this is presented in the next section of the paper.

4 System evaluation

The experiments that were carried out involve gender information retrieval, co-reference and anaphora resolution and finally summarization. In the following, each of these experiments are presented, highlighting the data set used, the systems selected for result comparison and the outcome.

4.1 Gender Information Retrieval

Gender related information was extracted from two GATE resource files: *person_male* and *person_female* gazetteers. For evaluation we manually annotated 15 random documents taken from the Reuters RCV1 [9] data set. The two systems that were compared with the manually obtained results are:

- Our system, henceforward referred to as *System*
- A *Baseline* system, which assigns the *masculine* gender to all named entities labelled as persons.

The results are presented in Table 4.1.

	Masculine	Feminine	Total
System	170/206 (83%)	7/14 (50%)	177/220 (80%)
Baseline	206/206 (100%)	0/14 (0%)	206/220 (94%)

Table 4.1: Gender evaluation results.

The fact that *System* correctly labelled a significant percent of masculine as well as feminine persons shows it will carry out gender retrieval better than the baseline system when the number of persons belonging to either genders will be more balanced.

4.2 Co-reference Resolution

For the evaluation of co-reference resolution the same set of 15 articles mentioned in section 4.1 was used. Named entities were extracted based on GATE, and the co-reference resolution performed by *System* was compared with the one of GATE. The results are shown in Table 4.2. There are 783 named entities extracted using GATE. The *System* performance is better than that of GATE, 750 entities compared to GATE's 646 entities co-referenced.

	Co-References
System	750/783 (96%)
GATE	646/783 (83%)

Table 4.2: Co-reference evaluation results.

4.3 Anaphora Resolution

In the case of anaphora resolution, the *System* was compared with two baseline systems. Both of them consider the closest named entity as a pronoun replacement, but one takes gender information into account, whereas the other does not. In this case, we randomly chose a subset of the Reuters data set formed of 77 articles.

Pronouns	System	Baseline-gender	Baseline-no gender
He	35/42 (83%)	18/42 (43%)	18/42 (43%)
They	7/20 (35%)	8/20 (40%)	2/20 (10%)
I	4/15 (27%)	0/15 (0%)	2/15 (13%)
She	0/0	0/0	0/0
Who	0/0	0/0	0/0
It	11/35 (31%)	11/35 (31%)	11/35 (31%)
Other	2/4 (50%)	2/6 (33%)	3/6 (50%)
Total	**59/116 (51%)**	**39/118 (33%)**	**36/118 (31%)**

Table 4.3: Anaphora evaluation results.

The results are listed in Table 4.3, pointing out the *System* strength where the "*he*" pronoun is concerned.

4.4 Summary Generation

For summarization evaluation, two tests were carried out. The first one involved the usage of the DUC (*Document Understanding Conferences*) [10] 2002 data set, for which the results obtained were similar with the ones listed in [4] . For the second one the DUC 2007 update task data set was used for testing purposes. The data consisted of 10 topics (A-J), each divided in 3 clusters (A-C), each cluster with 7-10 articles. For this assessment, we focused on the first part of the task – *producing a summary of documents in cluster A* – 100-words in length, without taking into consideration the topic information. In order to obtain the 100-word summary, we first retrieved all sentences having triplets belonging to instances with the class attribute value equal to +1, and ordered them in an increasing manner, based on the value returned by the SVM classifier. Out of these sentences, we considered the top 15%, and used them to generate a summary. That is because most sentences that were manually labelled as belonging to the summary were among the first 15% top sentences.

In order to compare the performance of various systems, we employed ROUGE (*Recall-Oriented Understudy for Gisting Evaluation*) [11] , an automatic summarization evaluation package. Our system was ranked 17 out of 25, based on the ROUGE-2 evaluation method, and 18 out of 25 based on the ROUGE-SU4 evaluation method (Figure 4.1 and Figure 4.2).

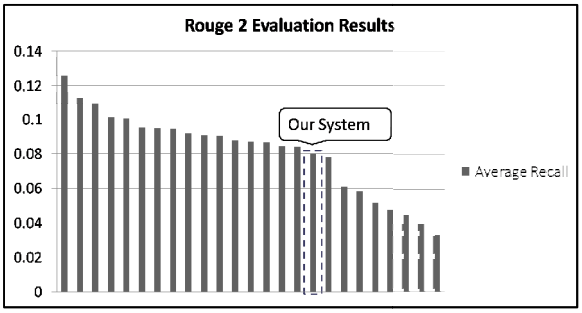

Figure 4.1: ROUGE-2 average recall results for 25 systems.

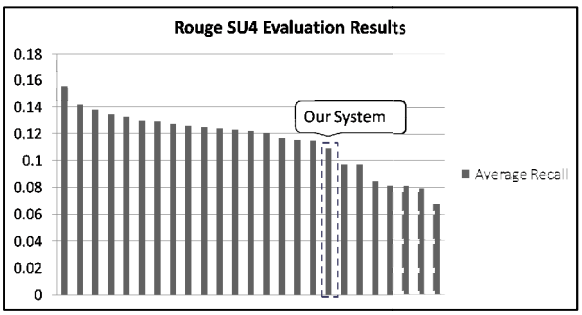

Figure 4.2: ROUGE-SU4 average recall results for 25 systems.

5 Conclusion

The stated purpose of the paper was to present a methodology for generating semantic graphs derived from logical form triplets and, furthermore, to use these semantic graphs to construct document summaries. The evaluation that was carried out showed the system in comparison to other similar applications, demonstrating its feasibility as a semantic graph generator and document summarizer.

As far as future improvements are concerned, one possibility would be to combine the document summarizer with an online newswire crawling system that processes news on the fly, as they are posted, and then uses the summarizer to obtain a compressed version of the initial story.

6 References

[1] D. Rusu, L. Dali, B. Fortuna, M. Grobelnik, D. Mladenić. Triplet Extraction from Sentences. Ljubljana: 2007. Proceedings of the 10th International Multiconference "Information Society - IS 2007". Vol. A, pp. 218 - 222.

[2] Stanford Parser: http://nlp.stanford.edu/software/-lex-parser.shtml

[3] GATE (General Architecture for Text Engineering): http://gate.ac.uk/

[4] J. Lescovec, M. Grobelnik, N. Milic-Frayling. Learning Sub-structures of Document Semantic Graphs for Document Summarization .Seattle: 2004. KDD 2004 Workshop on Link Analysis and Group Detection (LinkKDD).

[5] R. Mitkov. Robust pronoun resolution with limited knowledge. Montreal: 1998. Proceedings of the 18th International Conference on Computational Linguistics COLING'98/ACL'98. pp. 869-875.

[6] R. Mitkov. Anaphora Resolution: The State of the Art. Wolverhampton: 1999. Working Paper (Based on the COLING'98/ACL'98 tutorial on anaphora resolution).

[7] J. M. Kleinberg. Authoritative Sources in a Hyperlinked Environment. 1998. Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms.

[8] L. Page, S. Brin, R. Motwani, T. Winograd. The PageRank Citation Ranking: Bringing Order to the Web.1998.

[9] D. D. Lewis, Y. Yang, T. G. Rose, F. Li. RCV1: A New Benchmark Collection for Text Categorization Research. 2004, Journal of Machine Learning Research, Vol. 5.

[10] DUC (Document Understanding Conferences): http://duc.nist.gov/

[11] ROUGE (Recall-Oriented Understudy for Gisting Evaluation): http://haydn.isi.edu/ROUGE/