Enriched semantic graphs for extractive text summarization

Antonio F. G. Sevilla, Alberto Fernández-Isabel, and Alberto Díaz

Department of Software Engineering and Artificial Intelligence
Universidad Complutense de Madrid, Madrid, Spain,
afgs@ucm.es, afernandezisabel@ucm.es, albertodiaz@fdi.ucm.es

Abstract. Automatic extraction of semantic information from unstructured text has always been an important goal of natural language processing. While the best structure for semantic information is still undecided, graph-based representations enjoy a healthy following. Some of these representations are extracted directly from the text and external knowledge, while others are built from linguistic insight, created from the deep analysis of the surface text. In this document a combination of both approaches is outlined, and its application for extractive text summarization is described. A pipeline for this task has been implemented, and its results evaluated against a collection of documents from the DUC2003 competition. Graph construction is fully automatic, and summary creation is based on the clustering of conceptual nodes. Different configurations for the semantic graphs are used and compared, and their fitness for the task discussed.

Keywords: Semantic graph, information extraction, text summarization, natural language processing

1 Introduction

Nowadays, information accessible through the Internet is always increasing. However, it is often present in the form of unstructured text. Applications that want to exploit this information have to face multiple problems related to knowledge extraction, and find a method to understand the true meaning of the data.

Either in a way parallel to human language understanding, or with alternative methods better suited for machine processing, the purpose of these applications is often to transform a text into a representation which can be further utilized. One of these representations is the conceptual graph. This graph is a semantic network which links the concepts present in the text using the relations that are deduced from it.

There are many alternatives for building a semantic graph. Some of them are focused on using the nouns in the sentences [12], while other proposals use verbs and other content words (e.g. adverbs or adjectives) [9] for representing the main concepts of the target text. This approach is closer to syntactical analysis and can often offer a richer structure. In both cases, concepts correspond to the nodes in the graph, while the links between them represent the implicit semantic information of the text.

In this document, a fully automatic process is described based on conceptual graphs. They are built from a previous dependency analysis of English text by Freeling [11]. These graphs are enriched with information based on both linguistic analysis [14] and external lexical knowledge. This knowledge extends the node structure with lexical information recovered from WordNet [8]. This provides a hypernym hierarchy which bridges related concepts from different sentences or paragraphs. The main entities in the graph are the concepts, nominal or otherwise, while the links are not limited to grammatical relations. Concept nodes are also linked to each other based on semantic similarity, in order to add implicit knowledge to the graph. This pipeline is implemented with an in-development conceptual graph library, code-named "Grafeno", which is to be released as open-source.

In this paper, the semantic graphs generated are used to perform text summarization, but the developed library is intended to be general enough. It can be used to create and manipulate concept graphs in different ways, and information (such as triplets, a description of relations as 3-tuples $relation(concept_1, concept_2)$ [13]) can be extracted from it in text.

For the implemented summarization pipeline, clustering of the nodes based on a degree-based method [2] is performed, in order to identify the main topics of the target document. Text compression is achieved through an extractive method, where only a few sentences from the original document are selected. A heuristic based on selecting only the sentences related to the biggest cluster (main topic) is implemented [12].

For evaluation, eight documents from the DUC 2003 competition collection have been used. Summaries obtained with the extractive method are compared to a human-written summary, in order to measure the appropriateness of the approach.

The rest of the document is structured as follows: Section 2 situates the approach in the domain. Section 3 describes our semantic graph approach, its construction process where it is enriched with linguistic data, and the summarization process. Section 4 evaluates our proposal and discusses the results obtained. Finally, Section 5 discusses our conclusions and possible lines of future work.

2 Related work

There are multiple forms of representing the semantic content of documents and extracting this information. One of the most common is related to conceptual graphs [5], where the topics of the text are identified and organised, keeping its general meaning linking them with edges.

There is not a standard approach to the structure of semantic graphs. One can situate the main verb of the target sentence as the root of the sub-graph that represent it, while the child nodes describe the nouns and other complements [9]. These graphs are often based on syntactical analysis and dependency identification, and store linguistic information and structures [14]. In contrast, others approaches use only the nouns in the sentence [12]. The first case is the one more closely followed by our proposal.

For building rich semantic graphs, syntactic analysis is needed. This process can be automatized using different tools. One of these tools is Freeling [11], which performs many different layers of linguistic analysis and can be easily integrated into a bigger pipeline.

Other automatic tools support lexical analysis and semantic information retrieval, for enriching the graphs with data not present in the text. WordNet [8] is an on-line knowledge base where related terms can be obtained for nouns, verbs, adjectives or adverbs. It also provides different types of lexical relations, such as *synonymy* or *hypernymy* among others.

Regarding automatic text summarization, its goal is to preserve the main topics of a document while reducing its complexity and size [10]. It presents two different approaches according to the sources from which the final text is obtained: *extraction* and *abstraction*. The former builds the summary from the sentences of the original text, linking them to the topics identified in the semantic graph [16]. The latter generates text using external resources, which leads to new sentences created from the semantic concepts extracted. These sentences preserve the original meaning and ideally avoid any loss of information [3].

There are various metrics that can be used to measure automatic summaries made by software. In this approach, one of the ROUGE metrics is used (Recall-Oriented Understudy for Gisting Evaluation) [6]. It compares the frequency of overlapping words in the human and the automatic summaries, and obtains a value from 0 to 1.

3 Methodology

3.1 Overview

Graphs are, in essence, a set of nodes and the edges between them. In our case, the main information that nodes need to represent is a concept. Concepts are defined as lexemes, and represented by its lemma. Nodes can store additional information, in a free-form set of attribute-value pairs associated to the particular concept. These attributes can capture surface information with semantic meaning, like tense or gender, or can be used to store information obtained from external sources (e.g. a WordNet synset name).

Linking the nodes in the graph there are edges, which connect pairs of concepts. Edges are directed, meaning there is a head or parent node and a tail or child one. The main information that edges represent is the concepts that they link, but also how they link them and what relation do they represent.

These relations can be of various types. Semantic ones associate verbal concepts with their arguments, like the agent or the theme of a predicative verb,

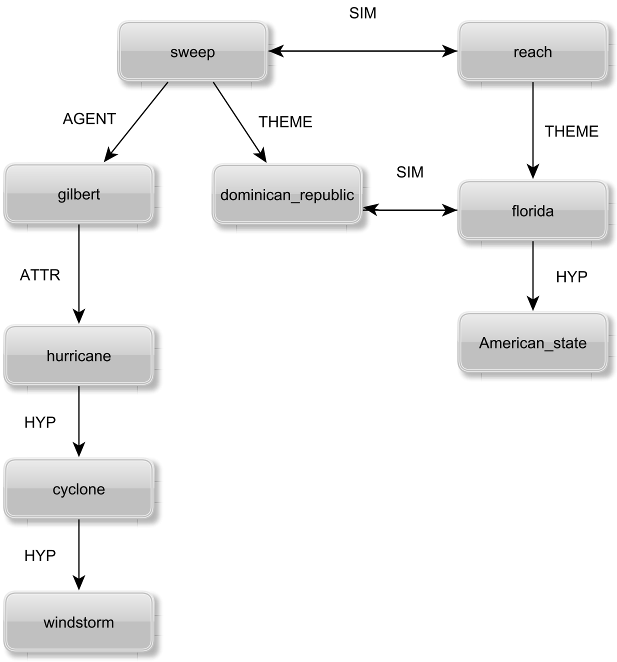

Fig. 1. Example extracted semantic graph for the text: *"Hurricane Gilbert swept the Dominican Republic Sunday evening. It then reached Florida."*

or nominal entities with their modifying attributes. Discourse relations mark the succession of predicates in the text, and can potentially link anaphora to their antecedent. Knowledge relations link concepts with similar ones in terms of an external knowledge base (e.g. hypernyms and hyponyms from WordNet, or entailed predicates from ConceptNet [7]).

And just as nodes can have attributes, edges can have them too. These attributes serve to further specify the information conveyed by the relation. For instance, a complement relation to a verb can have an attribute discriminating duration, location, direction, or other types of adverbial modification. Additionally, the attributes can be used to store the confidence or relevance of the relation, in the cases where the original source of information is empirical data or heuristic operations.

An example extracted graph can be seen in Fig. 1. It features semantic edges, like `AGENT` and `THEME`, or `ATTR` for attribute. `SIM` edges link similar concepts, and `HYP` edges bring in the hypernym hierarchy from WordNet.

3.2 Enriched Semantic Graphs

The process of building the semantic graph must take a free-form text in the source language (in this case English), and output a conceptual representation of its meaning. The text input must already be pre-processed, in the sense of removing non-textual elements, such as html tags and other formatting structure, only the text proper remaining.

Before creating our deep-level representation of the text, the lower levels must be analyzed. We use the Freeling tool to perform these tasks. In particular, we ask the tool to extract a dependency representation of every sentence. In order to do this, the tool performs all previous steps of tokenization, phrase structure parsing, and even named entity recognition. The resultant parse is a dependency tree, with the words and their syntactic information as nodes.

After the syntax tree is obtained, it is transformed into a semantic graph. Different transformations can be used, ranging from simple extraction of content words based on part of speech information, to more complex rules that understand the different dependencies. Concepts are identified and added to the graph as nodes, and links between them are then found. These links can be based on syntactic information, such as adjective-noun modification or verbal arguments. Other links can also be added, relating concepts which are similar from an information-content point of view, or bridging anaphoric and co-referential nodes.

While the graph is being built, contextual information is also added to it. Lexical information is queried from WordNet, finding hypernyms for each term and linking them to their hyponyms. Since these hypernyms are added as nodes, but only once, they serve as a bridge between concepts which may appear far from each other in the text.

3.3 Using the semantic graph for text summarization

This graph-based semantic structure can then be used for different procedures, one of them being extractive text summarization [16]. To perform this task, two operations are required, and are presented below.

The first applies a clustering algorithm to the graph, with the purpose of identifying the main topics of the text. The second aligns the original sentences of the document to the clusters found, enabling the subsequent generation of the summary.

Clustering This operation finds different groups (clusters) in the graph, which represent the topics of the original text. The grouping is based on connectivity, using a degree-based algorithm [2].

Clusters are created around the main concepts, which are found as the centroids of highly connected subgraphs. This is based on the assumption that documents written in English build a free scale network [1], and as a consequence the graph of target text is a network of this type. These networks present a few nodes highly connected between them (*Hub nodes*), while the rest of nodes have a relatively low connectivity.

Regarding the algorithm used to find the clusters, the fist step consists of locating and grouping the most connected nodes of the graph using the salience attribute [15]. A number between 2% and 20% of them are selected, called the *Hub nodes.* The exact number does not affect much the main clusters that are found. Since in this approach only the biggest one is used, the proportion is irrelevant, and so is fixed for the experiments. These vertices are then grouped in *Hub Vertex Sets (HVS).* These are sets of nodes highly connected, and serve as the centroids of the clusters to be found. They are identified using the rule that the connectivity between the concepts of a cluster must be the highest, while the connectivity between different clusters should be minimal. Once the *HVS* are built, the last step involves linking the rest of nodes (i.e. those which are not *Hub nodes*) to the *HVS* to which they present most connectivity. Thus, the final clusters of nodes are obtained. Each cluster can be seen as a topic within the text, represented by the hub vertices in them.

Sentence selection After the clusters are identified, a score is computed for each sentence and cluster. The idea is to rank sentences as related to each topic, in order to extract the most relevant ones.

The scoring algorithm uses a voting mechanism [15]. During graph construction, the nodes corresponding to each sentence have been recorded. After clustering is performed, these nodes emit a vote for the cluster they belong to. This vote is qualified, counting double (i.e. 1 instead of 0.5) for those nodes belonging to the hub vertex set of each cluster.

Votes are then added, producing a function that for each sentence and cluster, gives a measure of how related the former is to the latter. This measure can be used to rank sentences as more or less relevant to each topic.

Finally, the most relevant sentences are extracted, until the desired length for the summary is reached.

4 Evaluation

Text summarization has been chosen to perform an evaluation on the extracted conceptual graphs. For this experiment, an approach similar to that in [12] in used, and eight documents from the DUC 2003 competition have been selected. These documents are news articles of variable length, for which a short summary of around a hundred words is provided. This summary has been written by a human, and serves as target of the evaluation.

The experiment uses as input the original article and creates a conceptual graph from the dependency analysis achieved by Freeling. Then, the topics of the texts are identified, applying the clustering algorithm from Section 3.3. For sentence selection, only the biggest cluster is used, since it represents the main topic. The sentences which best represent it are extracted from the original document, and concatenated (in the original order) to create a summary of the same length as the human-made one.

To have a baseline to compare to, another pipeline has been created, which creates the summary with the first sentences from the original text. For news

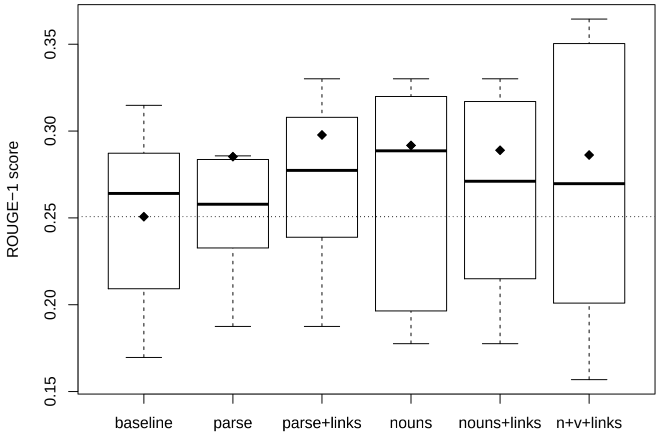

Fig. 2. Different graph configurations and their summary's Rouge scores. Boxplots show the median and interquartile range, and the diamonds mark the means. There is one outlier document, with scores of around 0.5, which is not shown.

articles this is a hard-to-beat baseline, since the most relevant information tends to appear at the beginning.

The created summaries are evaluated using the ROUGE recall-based metric, in particular ROUGE-N with unigram scores (in our case ROUGE-1) [6]. This metric can be used in same-length summaries, computing the overlap of n-grams between the evaluated text and the gold standard. The values range from 0 (no overlap) to 1 (exact same sequences of words). The results are illustrated in Fig. 2.

Since the pipeline for graph construction is fully customizable, different configurations have been tried. All of them extract concepts from the text, and also add to them the lexical hierarchy queried from WordNet. Their main differences lie in what concepts are used as nodes and what links are then added between them.

The 'parse' experiment constructs the graph using all content words, and adds the semantic relations extracted from the dependency parse. 'parse+links' also links similar concepts according to the Jiang-Conrath Similarity measure [4]. 'nouns' does not use any sentence structure information, and just finds nominal entities in the text. 'nouns+links' again only uses nouns, but also adds the previously mentioned similarity links. 'n+v+links' is like the previous one, but also uses verbal entities as conceptual nodes.

4.1 Discussion of results

The original idea for these experiments was to find a way to evaluate the more expressive concept graphs that we have created. Building on previous work, we expected that, since our concept graphs were richer than those used in previous approaches, they would improve performance in already existing and successful summarization methods. The more information there is in a graph, the better the summary that will be created from it.

As can be seen in Fig. 2, our results do slightly improve on the baseline. A few other experiments have been tried, with very similar results. However, this improvement is not statistically significant, and extremely dependent on the source text. While this may seem as a negative result, it can be explained with two main arguments.

On the one hand, the evaluation summaries are written by humans. These documents represent better an abstractive summarization approach, since the human writers do not just copy sentences, but rather rewrite them. In fact, the outlier document where we achieve much higher score is one where the language used by the human summary is extremely close to that of the original text. But most summaries are not like this, and therefore ROUGE scores, which take into account the words used, have a very low ceiling. Since the words and phrasing that the human summarizer uses are not present in the original document, it is impossible for an extractive method to reproduce them.

On the other hand, topic extraction by clustering does not appear to improve when the graphs are enriched by additional semantic information. In our opinion, this is due to the fact that clustering performs better when the graph is homogeneous. Thus, a graph composed solely of nominal entities will yield better clusters than one which includes verbs, adjectives or adverbs, and in which edges represent information of very heterogeneous nature.

On top of this, the loss of information from only using nouns does not actually hurt the extractive summarization method. Nouns represent the entities the text is talking about, and doing clustering on them accurately finds the main topics. And while the key information of what is being said about the topics is not present in the graph, it is put back by the extractive summarizer when selecting sentences from the original text.

This only works, however, when doing extractive summarization. If the conceptual graphs are to be used for further processing, leaving out all information but that of the entities present is unacceptable. This is precisely what we are trying to solve with our graphs, which include as much semantic information as possible.

Moreover, the enriched conceptual graphs manage to reach similar scores as those without the semantic enrichment. This means that they provide at least the same information as the previous ones, so what is being added is not noise. It just so happens that the additional data does not seem to improve extractive summarization via clustering.

5 Conclusions

This document has introduced a graph-based approach to extracting semantic information from text. It proposes a fully automatic graph building process, based on the result generated by a language processing tool such as Freeling. This tool takes a free-form text and generates a dependency parse, from which then the conceptual graph is created. Nodes in the graph represent concepts present or related to the text, and the edges describe their relations.

The graphs can then be clustered, and subgraphs extracted where the most important topics of the text are identified. These topics can then be used to score sentences for generating a short summary.

While we remain interested in using our richer concept graphs for semantic representations of texts, extractive summarization seems not to be the most appropriate application for them. Results are not conclusive, but exploration seems to indicate that there is not much room for improvement. An abstractive approach would probably be a better fit, and this leads to a possible line of future work in natural language generation based on concept graphs.

Another line of future work lies in creativity, an abstract step before this possible language generation. In this line, conceptual graphs can be merged or explored to find new pieces of information not present in the original text. This information can be deduced from the context added to the graph, or may be even created by observing and following patterns in the data.

Alongside further applications of concept graphs for processing, a clear line of work lies in the continuation of the enrichment effort. More semantic links can be found between the different concepts, understanding more complex dependency relations. Some function words that are now dropped as non-semantic, like determiners, can be interpreted to add nuances to the concepts found. Since the underlying dependency parser is able to give us a very complete description of the sentence, there is much information still waiting to be exploited.

In the same line of better linguistic enrichment, even more sophisticated relations can be found. Algorithms for endophora or even exophora resolution can be implemented, adding nodes and links, or merging co-referential nodes. Since the conceptual graph is not tied to the sentence or even the text structure, transformations can be performed to better convey the true meaning of the text. Other pragmatic relations can be found: discourse structure can be added between the different statements, or information structure in the sentence extracted. In this line, maybe clustering over the sub-graph of nominal nodes, as we already do for summaries, can be found to be a novel way of separating the topic and the focus of each sentence.

Acknowledgements

This work is funded by ConCreTe. The project ConCreTe acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET grant number 611733.

This research is funded by the Spanish Ministry of Economy and Competitiveness and the European Regional Development Fund (TIN2015-66655-R (MINECO/FEDER)).

References

1. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. science **286**(5439), 509–512 (1999)
2. Erkan, G., Radev, D.R.: Lexrank: Graph-based lexical centrality as salience in text summarization. Journal of Artificial Intelligence Research pp. 457–479 (2004)
3. Fiszman, M., Rindflesch, T.C., Kilicoglu, H.: Abstraction summarization for managing the biomedical research literature. In: Proceedings of the HLT-NAACL workshop on computational lexical semantics, pp. 76–83. Association for Computational Linguistics (2004)
4. Jiang, J.J., Conrath, D.W.: Semantic similarity based on corpus statistics and lexical taxonomy. arXiv preprint cmp-lg/9709008 (1997)
5. Leskovec, J., Grobelnik, M., Milic-Frayling, N.: Learning semantic graph mapping for document summarization. In: Proceedings of ECML/PKDD-2004 Workshop on Knowledge Discovery and Ontologies (2004)
6. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Text summarization branches out: Proceedings of the ACL-04 workshop, vol. 8 (2004)
7. Liu, H., Singh, P.: Conceptnet: a practical commonsense reasoning toolkit. BT technology journal **22**(4), 211–226 (2004)
8. Miller, G.A.: Wordnet: a lexical database for english. Communications of the ACM **38**(11), 39–41 (1995)
9. Miranda, S., Gelbukh, A., Sidorov, G.: Generación de resúmenes por medio de síntesis de grafos conceptuales. Revista signos **47**(86), 463–485 (2014)
10. Moawad, I.F., Aref, M.: Semantic graph reduction approach for abstractive text summarization. In: Computer Engineering & Systems (ICCES), 2012 Seventh International Conference on, pp. 132–138. IEEE (2012)
11. Padró, L., Stanilovsky, E.: Freeling 3.0: Towards wider multilinguality. In: LREC2012 (2012)
12. Plaza, L., Díaz, A., Gervás, P.: Concept-graph based biomedical automatic summarization using ontologies. In: Proceedings of the 3rd textgraphs workshop on graph-based algorithms for natural language processing, pp. 53–56. Association for Computational Linguistics (2008)
13. Rusu, D., Fortuna, B., Grobelnik, M., Mladenić, D.: Semantic graphs derived from triplets with application in document summarization. Informatica **33**(3) (2009)
14. Sowa, J.F.: Conceptual structures: information processing in mind and machine (1983)
15. Yoo, I., Hu, X., Song, I.Y.: A coherent graph-based semantic clustering and summarization approach for biomedical literature and a new summarization evaluation method. BMC bioinformatics **8**(9), 1 (2007)
16. Zhang, P.Y., Li, C.H.: Automatic text summarization based on sentences clustering and extraction. In: Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on, pp. 167–170. IEEE (2009)