Enhanced Web Page Content Visualization with Firefox

Lorand Dali, Delia Rusu, and Dunja Mladenić

Jožef Stefan Institute, Jamova 39, 1000 Ljubljana
{lorand.dali,delia.rusu,dunja.mladenic}@ijs.si

Abstract. This paper aims at presenting how natural language processing and machine learning techniques can help the internet surfer to get a better overview of the pages he is reading. The proposed demo is a Firefox extension which can show a semantic graph of the text in the page that is currently loaded in the browser. The user can also get a summary of the web page she is looking at by choosing to display only the more important nodes in the semantic graph representation of the document, where importance of the nodes is obtained by machine learning techniques.

Keywords: natural language processing, document visualization, triplet ranking, Firefox extension.

1 Introduction

The popularity and size of the Internet has greatly increased lately. So has the time people spend surfing. Moreover, a significant part of the traditional media has been replaced by online content. While it is very good that information can be retrieved so quickly and easily, the increased amount of time needed to be spent online by the users to process all this information raises important health and sociological problems. We think that a tool which could render most of the text contained in web pages easier to grasp, would greatly improve the user experience, and decrease the time needed to get at least a first impression of the page content. In our opinion such a tool should have the following features:

- provide a graphical representation of text, as most people prefer pictures over text
- emphasize and display the most important content (e.g., text summary)
- the user should be able to adjust the amount of information he wants to see
- the user should be able to see the content of the summary in the order in which he would see it during a normal read of the page
- the tool should be available and easy to use for the most users on the internet, and be applicable to most of the web pages

We believe that the tool presented in this paper satisfies the abovementioned needs. It is implemented as an extension to one of the most popular browsers, Firefox, and is designed to be used with any web page which has textual content.

The screenshot in Fig. 1 shows a typical usage scenario. The user wants to read a news article online and with the help of our browser extension he can also get a graphical representation of the text in the article.

Fig. 1. Screenshot showing a web page and its visualization as a graph

The natural language processing and machine learning techniques used are exposed to the Firefox extension through RESTful web services and the visualization is done using the TouchGraph¹ graph drawing library. In the following sections we shall present the general architecture and the building blocks of the system in more detail.

2 System Architecture

In this section we describe the general architecture of our system, as a pipeline [1][2]. First the HTML code of the web page under consideration is preprocessed to obtain plain text which is split into sentences. For removing the HTML markup we use the Python HTML parsing library BeautifulSoup² whose main advantage is that it can handle badly formatted HTML, whereas splitting the text into sentences is done with the Natural Language Toolkit's³ sentence detector. After that named entities and triplets are extracted from the sentences. For extracting named entities (people, places and organizations) we use GATE⁴ (General Architecture for Text Engineering), while (*subject – verb – object*) triplets are extracted from the parse tree of each sentence

¹ <http://sourceforge.net/projects/touchgraph>

² <http://www.crummy.com/software/BeautifulSoup/>

³ <http://www.nltk.org/>

⁴ <http://gate.ac.uk/>

obtained with the OpenNLP⁵ parser, using heuristic rules described in [3]. Triplets are enhanced by linking them to co-referenced named entities (where appropriate), by resolving anaphora and by semantic normalization which means identifying the WordNet⁶ synset their elements belong to. Next triplets are merged, becoming the building blocks of the semantic graph, where the subject and object are nodes, and the link is labeled with the verb. The semantic graph size can be reduced by ranking the triplets with the aid of machine learning techniques. Fig. 2 shows a block diagram of what we described before. The arrows are the processing tasks and the blocks represent the state in the system pipeline.

Fig. 2. System overview

3 Triplet Ranking Using Machine Learning Techniques

Ranking those triplets which compose the semantic graph, in decreasing order of importance has the goal of reducing the size of the semantic graph while retaining the most important nodes. From the list of ranked triplets a smaller semantic graph of any chosen size can be built.

The ranking method tried consists of training an SVM model for binary classification of triplets into important and not important. The 69 features used are of three kinds, as already proposed in [1]: *document features* (e.g. position of the sentence in the document, position of the triplet in the sentence, words in the triplet elements), *linguistic features* (e.g. part of speech tags, location of the triplet in the parse tree) and *graph features* (e.g. hub and authority weights, page rank, node degrees, connected components). For training the linear SVM model and for evaluating the triplet ranking, we use the DUC (Document Understanding Conferences)⁷ datasets from 2002 and 2007, respectively. The DUC datasets contain news articles from various sources like Financial Times or Wall Street Journal. The 2002 dataset comprises 300 newspaper articles on 30 different topics and for each article we have a 100 word human written abstract. The DUC 2007 dataset comprises 250 articles for the update task and 1125 articles for the main task, part of the AQUAINT dataset⁸; the articles are grouped in clusters and 4 NIST assessors

⁵ <http://opennlp.sourceforge.net/>

⁶ <http://wordnet.princeton.edu/>

⁷ <http://duc.nist.gov/>

⁸ <http://www ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2008T25>

manually create summaries (of 100 or 250 words) for the documents in the clusters. As training data we used the DUC 2002 articles, as well as the DUC 2007 main task articles, while the DUC 2007 update task articles were used for testing. We extracted triplets from the training and test data, and learned to classify if a triplet will appear in the summary or not. If we order the classified triplets by the confidence weights of their class we obtain a ranked list of triplets which is exactly what we need. The approach was also used to automatically create document summaries [2]. It was evaluated by comparing the results to the ones obtained by other systems competing in the DUC 2007 update task. Other competing systems that were ranked close to ours generate compressed versions of source sentences as summary candidates and use weighted features of these candidates to construct summaries [4], or learn a log-linear sentence ranking model by maximizing three metrics of sentence goodness [5] (see [2] for more details on the evaluation outcome).

4 Conclusions

We have presented a tool which is meant to make the reading of web pages easier by providing interactive graphical visualization of the read document. To achieve this task we have used natural language processing and machine learning techniques such as: named entity recognition, triplet extraction, co-reference and anaphora resolution and triplet ranking. The tool is implemented as a Firefox extension, which makes it available to a wide range of users browsing a big variety of web sites.

Acknowledgements

This work was supported by the Slovenian Research Agency and the IST Programme of the EC under NeOn (IST-4-027595-IP) and PASCAL2 (IST-NoE-216886).

References

1. Leskovec, J., Grobelnik, M., Milic-Frayling, N.: Learning Sub-structures of Document Semantic Graphs for Document Summarization. In: Workshop on Link Analysis and Group Detection (LinkKDD) at KDD 2004, Seattle, USA, August 22 – 25 (2004)
2. Rusu, D., Fortuna, B., Grobelnik, M., Mladenić, D.: Semantic Graphs Derived From Triplets With Application In Document Summarization. *Informatica Journal* (2009)
3. Rusu, D., Dali, L., Fortuna, B., Grobelnik, M., Mladenić, D.: Triplet Extraction from Sentences. In: Proceedings of the 10th International Multiconference "Information Society - IS 2007", Ljubljana, Slovenia, October 8 – 12, pp. 218–222 (2007)
4. Madnani, N., Zajic, D., Dorr, B., Ayan, N.F., Lin, J.: Multiple Alternative Sentence Compressions for Automatic Text Summarization. In: Proceedings of the Document Understanding Conference, DUC (2007)
5. Toutanova, K., Brockett, C., Gamon, M., Jagarlamudi, J., Suzuki, H., Vanderwende, L.: The PYTHY Summarization System: Microsoft Research at DUC 2007. In: Proceedings of the Document Understanding Conference, DUC (2007)