In-browser Text Importance Detector
BSP - Phase I deliverable

Sunday 3rd April, 2022 - 16:46

Paulo Ricardo Botelho Oliva
University of Luxembourg
Email: paulo.botelho.001@student.uni.lu

This deliverable has been produced under the supervision of:
Luis Leiva
University of Luxembourg
Email: luis.leiva@uni.lu

1. Project main objectives

This Bachelor Semester Project is all about finding important sentences in a text. The main objective of the project is to create a browser extension that highlights the most important information of an article's web page. Moreover, this project also addresses the topic of automatic text summarization, as well as extractive text summarization algorithms.

2. Prerequisites

The only required competency before starting to work on this Bachelor Semester Project is possessing enough knowledge about Web Technologies, such as HTML and the JavaScript programming language.

3. Scientific Deliverable Description

This BSP's scientific deliverable aims to answer the following research question:

How can we detect important sentences in a web document?

To help answer this question, we will need to answer the following secondary questions:

1. How can we automatically detect text blocks on a web page?

2. How can we extract relevant sentences within a text block?

To answer our main scientific question, we will investigate the concept of automatic text summarization. Automatic text summarization is the technique of creating a shorter version of a text using a computer program. This shorter version of the text (i.e., the summary) contains the most relevant information of the original text.

Automatic text summarization methods are very much needed to deal with the ever-increasing volume of text data available online, both to help identify relevant information and to consume relevant information more quickly. There are many reasons why one would need automatic text summarization. For example, summaries reduce reading time and should be less biased than summaries made by a human person [1].

In general, there are two main kinds of text summarization techniques, extractive text summarization and abstractive text summarization. Extractive text summarization consists in selecting the most important parts of a text and extracting them to create a shorter summary of the same text. Abstractive text summarization consists in gathering the main ideas of a text and using those abstract ideas to generate a summary of the text, where the summary isn't just a copy of the old text, but actual new sentences.

Given the scope of this BSP, our scientific deliverable will focus on extractive text summarization. There exist many approaches to extractive text summarization, like statistical approaches, lexical chain based approaches, graph-based approaches, cluster-based approaches, and fuzzy logic based approaches, among others.

We will focus on graph-based text summarization, as it is commonly used by many algorithms to classify sentences by their importance on the source text. For example, Google's PageRank is a graph-based ranking algorithm used by Google to rank web pages [2]. This algorithm is the basis of other well-known graph-based text summarization algorithms like TextRank [3] and LexRank [4].

The graph-based method of text summarization is an unsupervised technique in which sentences or words are scored using a graph, therefore their convenience for this BSP. In a nutshell, the basic goal of graph-based methods is to extract the most relevant sentences from a block of text. Thus, the scientific deliverable will focus on explaining how graph-based

text summarization algorithms work.

To answer the secondary questions, the notion of text blocks should be explained. In the context of web pages, a text block is considered a group of paragraphs. Critically, we need to identify the most important text blocks in a web page (the main content) as well as unimportant text blocks, such as texts in the header or the footer of the page. Unimportant texts should not be considered for summarization.

A key challenge is that automatic detection of the main text blocks on a web page is not reliable, since every web page has a different structure and layout. Thus, detecting blocks of text is a hard task. The scientific deliverable of this BSP will study this question more thoroughly. As there does not seem to be a trivial solution, we will try to find the best compromise solution.

4. Technical Deliverable Description

The technical deliverable of this project focuses on creating a browser extension that detects the most important sentences on a specific web page and highlights them, so that the user only needs to read the highlighted sentences instead of the whole web page.

A browser extension is a small software package that adds features and functions to a browser. Extensions are built using well-known web technologies such as HTML, CSS, and JavaScript. Extensions rely on their own set of APIs, which are browser-dependent. For example, they allow developers to change the browser's default behaviours or inject custom stylesheets of JavaScript content before a web page is loaded.

The extension should be built using JavaScript and should work on the Google Chrome browser, given its large market share of 65-70% [5] [6], without breaking the functionality of existing websites. The extension will automatically highlight the most important sentences of an article or blog post in a specific web page.

In short, the goal of this technical work is to implement text summarization techniques in a browser environment, while creating a simple extension that is easy to use and that works on most websites.

References

[1] J.-M. Torres-Moreno, *Automatic text summarization*. John Wiley & Sons, 2014.

[2] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web." Stanford InfoLab, Tech. Rep., 1999.

[3] R. Mihalcea and P. Tarau, "Textrank: Bringing order into text," in *Proceedings of the 2004 conference on empirical methods in natural language processing*, 2004, pp. 404–411.

[4] G. Erkan and D. R. Radev, "Lexrank: Graph-based lexical centrality as salience in text summarization," *Journal of artificial intelligence research*, vol. 22, pp. 457–479, 2004.

[5] NetMarketShare, "Browser market share," 2020. [Online]. Available: https://netmarketshare.com/browser-market-share.aspx

[6] StatCounter Global Stats, "StatCounter Global Stats - Browser, OS, Search Engine including Mobile Usage Share," 2022. [Online]. Available: https://gs.statcounter.com/

[7] BiCS(2021), "BiCS Semester Projects Reference Document," University of Luxembourg, Tech. Rep., 2021.

[8] ——, "BiCS Bachelor Semester Project Report Template," University of Luxembourg, BiCS - Bachelor in Computer Science, Tech. Rep., 2021. [Online]. Available: https://github.com/nicolasguelfi/lu.uni.course.bics.global