In-browser Text Importance Detector

Sunday 15th May, 2022 - 18:35

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

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

Abstract

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

1. Introduction

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. Project description

2.1. Domains

2.1.1. Scientific. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.

2.1.2. Technical. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.

2.2. Targeted Deliverables

2.2.1. Scientific deliverable. 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.

2.2.2. Technical deliverable. 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.

3. 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.

3.1. Scientific prerequisites

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris.

3.2. Technical prerequisites

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris.

4. Scientific Deliverable

4.1. Requirements

The scientific deliverable produced in this project consists in answering the question "How can we detect important sentences in a web document?".

4.2. Design

4.3. Production

4.4. Assessment

5. Technical Deliverable

5.1. Requirements

The requirements are the characteristics that are required of the completed technical deliverable. The software developed for the technical deliverable must meet a number of functional and non-functional requirements.

5.1.1. Functional Requirements.

- Gathering the text on a web page.
- Generating a summary of the gathered text.
- Highlighting the important text on a web page.
- Running at least on Google Chrome.

Gathering the text

The browser extension must be able to gather the text of any web page. The text is gathered by the browser extension by using.

Generating a summary

The extension must allow the user to generate a summary of the text gathered from the web page.

Highlighting the important text

The extension must be able to highlight the important text on the web page, based on the summary generated by the summarization algorithm.

5.1.2. Non-functional Requirements.

- Providing a high-quality summary.
- Maintaining the functionality of existing websites.

Running on Google Chrome

The browser extension must be able to run on Google Chrome.

5.2. Design

5.3. Production

5.4. Assessment

Acknowledgment

The author would like to thank the BiCS management and education team for the amazing work done.

6. Conclusion

7. Plagiarism statement

I declare that I am aware of the following facts:

- As a student at the University of Luxembourg I must respect the rules of intellectual honesty, in particular not to resort to plagiarism, fraud or any other method that is illegal or contrary to scientific integrity.
- My report will be checked for plagiarism and if the plagiarism check is positive, an internal procedure will be started by my tutor. I am advised to request a pre-check by my tutor to avoid any issue.
- As declared in the assessment procedure of the University of Luxembourg, plagiarism is committed whenever the source of information used in an assignment, research report, paper or otherwise published/circulated piece of work is not properly acknowledged. In other words, plagiarism is the passing off as one's own the words, ideas or work of another person, without attribution to the author. The omission of such proper acknowledgement amounts to claiming authorship for the work of another person. Plagiarism is committed regardless of the language of the original work used. Plagiarism can be deliberate or accidental. Instances of plagiarism include, but are not limited to:
  1) Not putting quotation marks around a quote from another person's work
  2) Pretending to paraphrase while in fact quoting
  3) Citing incorrectly or incompletely
  4) Failing to cite the source of a quoted or paraphrased work
  5) Copying/reproducing sections of another person's work without acknowledging the source
  6) Paraphrasing another person's work without acknowledging the source
  7) Having another person write/author a work for oneself and submitting/publishing it (with permission, with or without compensation) in one's own name ('ghost-writing')
  8) Using another person's unpublished work without attribution and permission ('stealing')
  9) Presenting a piece of work as one's own that contains a high proportion of quoted/copied or paraphrased text (images, graphs, etc.), even if adequately referenced

  Auto- or self-plagiarism, that is the reproduction of (portions of a) text previously written by the author without citing that text, i.e. passing previously authored text as new, may be regarded as fraud if deemed sufficiently severe.

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 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

8. Appendix