An HCI Approach to Extractive Text Summarization: Selecting Key Sentences Based on User Copy Operations

Ilan Kirsh[1][0000−0003−0130−8691] and Mike Joy[2][0000−0001−9826−5928]

[1] The Academic College of Tel Aviv-Yaffo, Tel Aviv, Israel, kirsh@mta.ac.il
[2] University of Warwick, Coventry, United Kingdom, M.S.Joy@warwick.ac.uk

Abstract. Automatic text summarization is a very complex problem. Despite being intensively researched, automatic summaries are still considered to be of lower quality than manual summaries. This paper introduces a novel HCI approach to web page summarization. The proposed Crowd-Copy Summarizer follows the extractive text summarization approach of summarizing by selecting sentences within the text. The selection is performed by examining how frequently users copy certain sentences to their clipboards, for their own purposes. The most frequently copied sentences are included in the summary. Results from an early experiment are promising, as key sentences, such as introductory sentences, definitions, and important highlights, are copied frequently. Consequently, the generated summaries can provide good coverage of the main topics. This novel text summarization approach combines the best of both worlds: summarization based on collective human wisdom, without the expensive burden of manual summarization work.

Keywords: Automatic Extractive Text Summarization · Clipboard · Copy and Paste · Website · Web Page · Document · Text · Crowd Wisdom

1 Introduction

The need for automatic text summarization becomes increasingly apparent as the amount of textual information available grows. Despite decades of extensive research, the quality of automatic summaries is still inadequate [8].

There are two main approaches to automatic text summarization: the extractive approach and the abstractive approach [2, 7, 8]. Extractive methods select key sentences from the text and compose a summary from these selected sentences, without changing them. Abstractive methods use Natural Language Processing (NLP) techniques to analyze the text and build a summary that may also contain synthetically generated sentences. Since abstractive summarization is very complex, abstractive summarization methods often rely on elements of extractive summarization [1].

Extractive methods usually calculate a score for every sentence and then select the sentences with the highest scores and include them in the summary.

In Proceedings of the 22nd HCI International Conference (HCII 2020), CCIS, Springer, 2020 (forthcoming).

Scores assigned to sentences are often based on scores given to individual words, as the importance of a sentence may be related to the importance of the words that it contains. Many different scoring methods have been studied [9]. The evaluation process usually combines information from the document (e.g. the frequency of a word in the document, where a higher frequency implies higher importance), with external knowledge (e.g. the frequency of a word in general, where a higher frequency implies lower importance) [7].

This paper introduces a new, "crowd wisdom" approach to text summarization of web pages. It follows the extractive approach (forming a summary by selecting important sentences within the text), only instead of selecting sentences using conventional methods, the new approach uses a novel source of information: copy operations of web users on web pages (for their own purposes). The most frequently copied sentences are included in the summary. To the best of our knowledge, this approach has never been studied before.

This paper is organized as follows. Section 2 describes the Crowd-Copy Summarizer implementation. Section 3 demonstrates the summarization of a sample web page. Section 4 analyzes the results. Section 5 concludes this paper.

2 Implementation

Figure 1 shows the architecture of the Crowd-Copy Summarizer. A reference to a *Copy Script* is embedded in all the web pages. As a result, every request for a page of the website returns a revised version of the page that triggers an additional request to load the *Copy Script* from the *Summarizer Server*. The script tracks JavaScript clipboard copy events and reports them back to the *Collector* component in the *Summarizer Server*. The *Collector* stores the data anonymized in a dedicated database, adhering to industry standards of data anonymization and user privacy preservation. The *Summarizer* uses the copy operations data and the original web page to produce the summary.

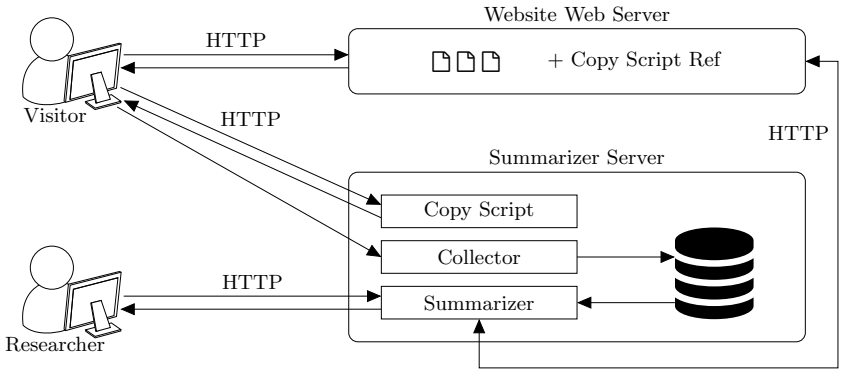

Fig. 1. High-Level Architecture of the Crowd-Copy Summarizer

Users copy strings of various types to the clipboard [4, 5], including, for example, words or sequences of words to look up or translate elsewhere [3, 6] and code fragments from code examples (copied by programmers to paste in their IDEs) [5, 6]. Therefore, the Summarizer ignores copy operations of the following types of content:

- content in a PRE HTML element, which usually contains code, and so is irrelevant in text summarization;
- content consisting of less than 8 words, which could be copied with the intention of searching for more information on the internet;
- content with more than 40 words, since including complete or large parts of paragraphs in summaries is likely to add noise.

The 8-40 words range was found to be reasonable through experimentation, but it is not necessarily optimal. The Summarizer breaks the text in every accepted copy operation into its constituent sentences and assigns one point to each complete sentence (partial sentences are ignored). The resulting summary is simply generated by joining all the sentences that exceed a minimum score threshold, ordered by their position in the original text.

3 Sample Summary

The Crowd-Copy Summarizer was tested on technical documentation web pages of the ObjectDB website[3]. This website contains learning materials on the Java Persistence API (JPA), the standard API for accessing databases from Java in an object-oriented way. Copy operations performed by visitors have been recorded for a period of three months, ending in March 2020.

Table 1 presents the text summarization of a sample page, which was viewed 3,847 times during that period. 312 copy operations have been recorded in total. 68 copy operations (consisting of 74 sentence occurrences) remained after applying the filtering process (as described in section 2). All the sentences with a score of 3 or above, ordered by their position in the page, are included in Table 1.

To provide a brief context, JPA refers to an ordinary object that represents data in the database as ‘managed’, and this sample web page introduces a different type of object, ‘detached’.

Different summaries of various lengths can be generated from the results in Table 1 by setting different minimum score thresholds. For example, setting the threshold to any value between 9 and 16 will generate a very short summary, consisting of a single sentence. This is the most important sentence, which defines the term detached, so it is probably the best possible one-sentence summary. This is the first indication that there is a positive correlation between the frequency of a sentence being copied and its importance.

[3] https://www.objectdb.com

Table 1. Summarization of the "Detached Entity Objects" Page

#	Sentence	Score
1	Detached entity objects are objects in a special state in which they are not managed by any EntityManager but still represent objects in the database.	16
2	Compared to managed entity objects, detached objects are limited in functionality.	7
3	Retrieval by navigation from detached objects is not supported, so only persistent fields that have been loaded before detachment should be used.	3
4	Changes to detached entity objects are not stored in the database unless modified detached objects are merged back into an EntityManager to become managed again.	8
5	Detached objects are useful in situations in which an EntityManager is not available and for transferring objects between different EntityManager instances.	4
6	When a managed entity object is serialized and then deserialized, the deserialized entity object (but not the original serialized object) is constructed as a detached entity object since is not associated with any EntityManager.	5
7	Marking a reference field with CascadeType.DETACH (or CascadeType.ALL, which includes DETACH) indicates that detach operations should be cascaded automatically to entity objects that are referenced by that field (multiple entity objects can be referenced by a collection field).	5
8	Detached objects can be attached to any EntityManager by using the merge method.	6
9	Marking a reference field with CascadeType.MERGE (or CascadeType.ALL, which includes MERGE) indicates that merge operations should be cascaded automatically to entity objects that are referenced by that field (multiple entity objects can be referenced by a collection field).	3

Page: https://www.objectdb.com/java/jpa/persistence/detach

Sentences are ordered by their appearance order in the page.

4 Analysis

We can expect a good summary to cover the most important information in the text, to eliminate redundancy, and to be readable. This section uses these three criteria in analyzing the summary that is produced from Table 1 by applying a score threshold of 3 (i.e. the summary that includes all the sentences in Table 1).

Most of the arguments in this discussion are applicable also to other thresholds (which produce shorter summaries).

4.1 Covering Important Information

The coverage of a summary can be assessed by examining if it answers the most important questions about the topic. In the context of a technical web page, the key questions about a new concept may be: What is it? When should we use it? How does it work? What are the differences between this new concept and other familiar concepts? We can see that most of the selected sentences in Table 1 answer these key questions:

- What are detached objects? Sentence #1 is the definition.
- How are they different from ordinary objects? Answered by #3 and #4.
- When are detached objects needed? Answered by #5.
- How do objects become detached? Answered by #6 and #7.
- How do objects stop being detached? Answered by #8 and #9.

These indeed seem to be the key questions. It seems that one important sentence is missing in Table 1: another part of the answer to the basic question of "How do objects become detached?" (by using the detach method). This sentence has not been selected as it was only copied once.

It is interesting to analyze the distribution of the copied sentences on the web page. This sample web page contains a preface and 5 sections. The 9 sentences in Table 1 are distributed as follows: 5 in the preface and one in each of sections 1, 2, 4, and 5. Section 3 (Bulk Detach) seems to be perceived as less important by the website users.

4.2 Avoiding Redundancy

The examined web page contains 6 headers, 26 sentences (9 of which are shown in Table 1), and 4 code boxes containing code fragments.

Examining sentences that were not copied by users (or rarely copied) shows that they discuss low-level details. For example, many sentences explain which exceptions are thrown when things go wrong, and these sentences are rarely copied by users. The general impression (to be verified in further work) is that sentences copied more frequently are indeed more important, and therefore, including them in the summary is justified.

One counter-example is sentence #2 in Table 1 that does not provide much value on its own. In fact, 6 out of the 7 copies that it scored were due to copy operations of both sentences #1 and #2 (which are adjacent in the text) combined. Counting only the first sentence in each copy operation may produce better summaries. In the summarization of this web page, it would only affect sentence #2: reducing its score from 7 to 1, removing it from Table 1, and eliminating it from any derived summary.

4.3 Preserving Readability

The resulting summary is quite readable (and so are other summaries produced from Table 1 using other thresholds). It seems that users tend to copy standalone sentences more frequently than sentences that depend on other sentences or code fragments (e.g. sentences that explain code). As a result, these self-contained sentences can be combined into a summary that does not feel fragmented.

5 Conclusions

This paper presents a new approach to extractive text summarization: composing a summary from sentences that are frequently copied by users.

Users copy to the clipboard strings of various types and for different purposes. Words and phrases are often copied to the clipboard in order to look them up on the internet. It is quite unlikely that complete sentences are copied for this purpose, as long strings are not effective in search. Full sentences may be copied in order to use them in summaries or as citations in documentations, presentations, blogs, websites, answers on forums (such as StackOverflow), or even in private communications between colleagues who work on a project together. Key sentences are probably copied more frequently, and therefore, the frequency of copying a sentence can be used in extractive text summarization as an indicator of its importance.

An initial analysis of the results is promising. Key sentences, such as introductory sentences, definitions, and important highlights, are copied more frequently. Consequently, summaries produced using this approach could provide good coverage of the main topics presented in web pages. Further work should include a full evaluation of this approach, including a comparison against conventional text summarization methods.

References

1. Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E.D., Gutierrez, J.B., Kochut, K.: Text summarization techniques: A brief survey. International Journal of Advanced Computer Science and Applications **8**(10) (2017). https://doi.org/10.14569/IJACSA.2017.081052, http://dx.doi.org/10.14569/IJACSA.2017.081052
2. Kiani, F., Tas, O.: A survey on automatic text summarization. Press Start **5**, 205–213 (06 2017). https://doi.org/10.17261/Pressacademia.2017.591, http://pressacademia.org/archives/pap/v5/29.pdf
3. Kirsh, I.: Automatic complex word identification using implicit feedback from user copy operations. In: Proceedings of the 21st International Conference on Web Information Systems Engineering (WISE 2020), Lecture Notes in Computer Science, forthcoming. Springer International Publishing, Cham (10 2020)
4. Kirsh, I.: What web users copy to the clipboard on a website: A case study. In: Proceedings of the 16th International Conference on Web Information Systems and Technologies (WEBIST 2020), forthcoming. INSTICC, SciTePress, Setúbal, Portugal (11 2020)

5. Kirsh, I., Joy, M.: A different web analytics perspective through copy to clipboard heatmaps. In: Proceedings of the 20th International Conference on Web Engineering (ICWE 2020), Lecture Notes in Computer Science, vol 12128. pp. 543–546. Springer International Publishing, Cham (06 2020). https://doi.org/10.1007/978-3-030-50578-3_41, https://doi.org/10.1007/978-3-030-50578-3_41
6. Kirsh, I., Joy, M.: Splitting the web analytics atom: From page metrics and KPIs to sub-page metrics and KPIs. In: Proceedings of the 10th International Conference on Web Intelligence, Mining and Semantics (WIMS 2020), Biarritz, France. pp. 33–43. Association for Computing Machinery, New York, NY, USA (06 2020). https://doi.org/10.1145/3405962.3405984, https://doi.org/10.1145/3405962.3405984
7. Rajasekaran, A., Varalakshmi, R.: Review on automatic text summarization. International Journal of Engineering and Technology(UAE) **7**, 456–460 (06 2018). https://doi.org/10.14419/ijet.v7i2.33.14210
8. Saggion, H., Poibeau, T.: Automatic Text Summarization: Past, Present and Future, pp. 3–13. Springer Berlin Heidelberg, Berlin, Heidelberg (01 2013). https://doi.org/10.1007/978-3-642-28569-1_1
9. Sajjan, R., Shinde, M.: A detail survey on automatic text summarization. International Journal of Computer Sciences and Engineering **7**, 991–998 (06 2019). https://doi.org/10.26438/ijcse/v7i6.991998