SUMMHELPER: Collaborative Human-Computer Summarization

Aviv Slobodkin[1]*, **Niv Nachum**[1]*, **Shmuel Amar**[1], **Ori Shapira**[2]†, **Ido Dagan**[1]

[1]Bar-Ilan University [2]Amazon
{lovodkin93, niv252, shmulikamar, obspp18}@gmail.com
dagan@cs.biu.ac.il

Abstract

Current approaches for text summarization are predominantly automatic, with rather limited space for human intervention and control over the process. In this paper, we introduce SUMMHELPER,[1] a 2-phase summarization assistant designed to foster human-machine collaboration. The initial phase involves content selection, where the system recommends potential content, allowing users to accept, modify, or introduce additional selections. The subsequent phase, content consolidation, involves SUMMHELPER generating a coherent summary from these selections, which users can then refine using visual mappings between the summary and the source text. Small-scale user studies reveal the effectiveness of our application, with participants being especially appreciative of the balance between automated guidance and opportunities for personal input.

1 Introduction

Text summarization is the task of generating a condensed version of a given text. Most summarization approaches operate in a fully automated pipeline. While efficient, fully automatic summarization does not flexibly enable human intervention and control during the summarization process, which could potentially tune the process to better accommodate user preferences, as well as rectify inevitable mistakes made by models. Our objective in this paper is to promote such a human-involved approach to summarization, allowing to better tailor the eventual output to real-world user needs, and to synergize the efficiency of the computer with the quality of the human (Hoc, 2000; Pacaux-Lemoine et al., 2017; Flemisch et al., 2019). The process can conveniently support a range of practical scenarios that require individual preferences, such as editors preparing summaries of articles, students condensing notes, or legal practitioners abridging contracts.

To advance such direction, we present SUMMHELPER, a 2-stage summarization assistant, which decomposes the summarization pipeline into two natural subtasks—content selection followed by summary generation—and facilitates human-machine cooperation in each subtask. On an input document, the process starts with the selection of content for the summary (§3.1). SUMMHELPER suggests possible salient content, efficiently pointing users to central information within the text (see [1] in Figure 1a). Users may accept or reject suggested spans, or highlight any other content to include in the summary ([3] in Figure 1a).

Upon receiving highlighted content within the text, SUMMHELPER subsequently consolidates it and generates a coherent summary (§3.2). This step coincides with the recently introduced Controlled Text Reduction task (CTR; Slobodkin et al., 2022), which produces a coherent fused version of the content of marked spans ("highlights") in a source document, as interpreted within the context of the full text. Once ready, users can review the generated summary and edit any unsatisfactory content. To facilitate inspection, users are presented with a side-by-side display of the summary and the highlighted input (see Figure 1b), with clearly marked alignments between summary spans and corresponding source text spans ([5] and [6] in Figure 1b). The automatic alignments aid users in navigating through the input text and identifying summary content that may need editing.

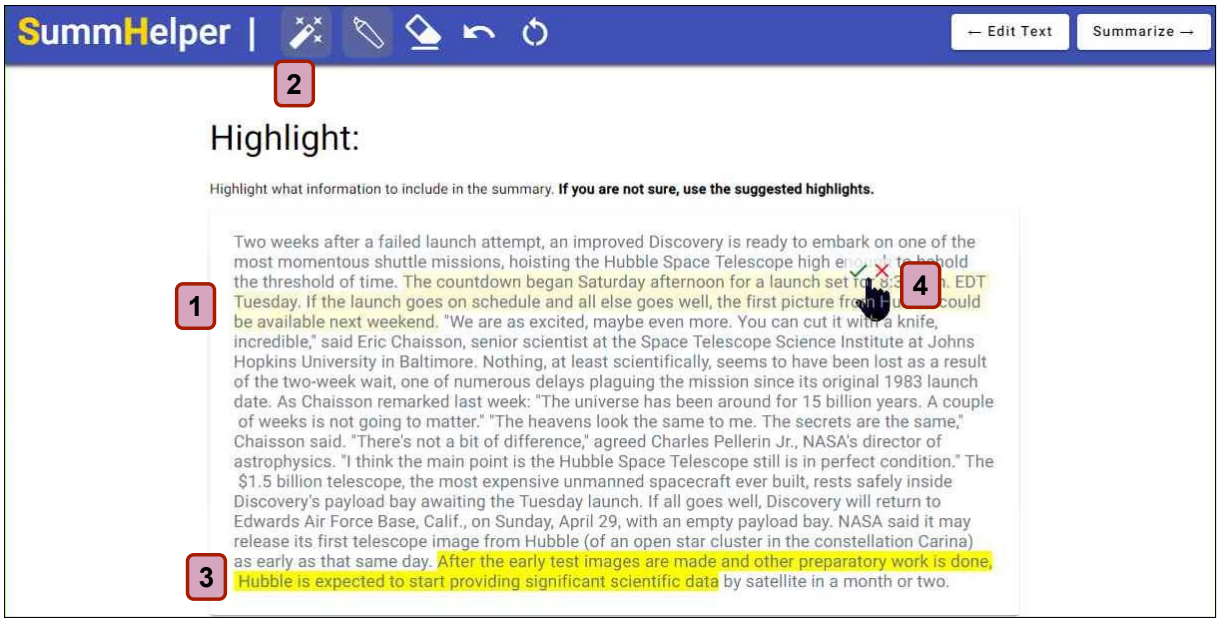
To assess SUMMHELPER's usefulness for generating customized summaries, we conduct two user studies (§4), following common human-computer

* Equal contribution.

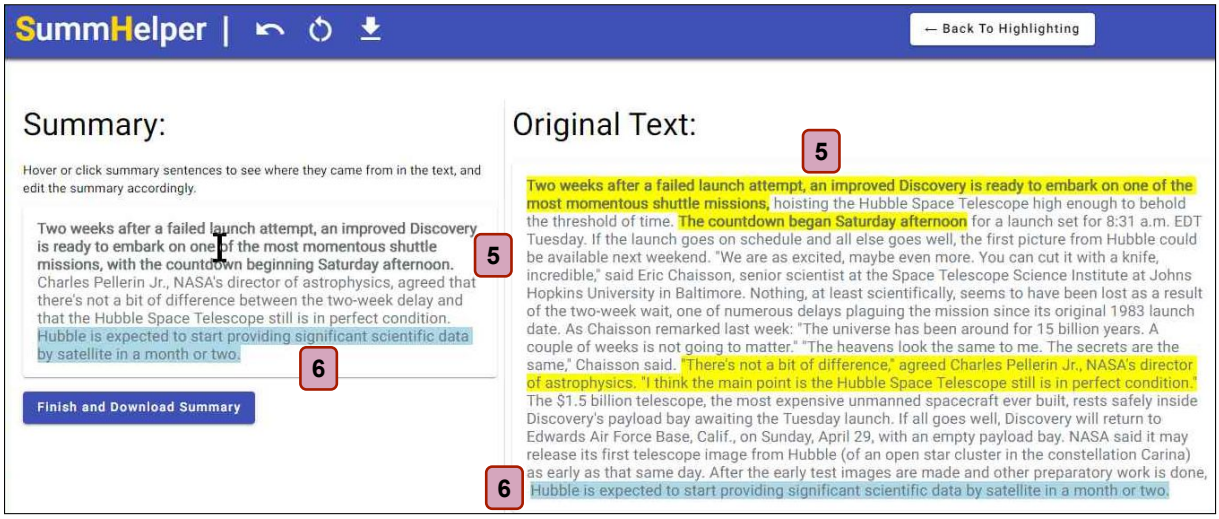
† Work done in cooperation with Bar-Ilan University (external and not related to the author's work at Amazon).

[1]System at https://nlp.biu.ac.il/~sloboda1/SummHelper, screencast demo at https://www.youtube.com/watch?v=jKzS9RwuccM and code is available at https://github.com/niv252/SummHelper

Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 554–565



(a) Content selection window



(b) Review and editing window

Figure 1: Our SUMMHELPER web application. First, users upload a document and enter the content selection window (1a) to select what information to include in the summary. Users can receive suggestions from the system (pale yellow; [1]), through the magic wand icon [2]. Any part in the text can be highlighted via mouse click-and-drag operations [3]. Users can also accept or reject entire suggested spans via the respective ✓ and ✗ buttons, which appear when hovering over suggestions [4]. When finishing highlighting, a summary is generated, and users proceed to the reviewing window (1b), which shows the generated summary and the source text, with highlights, side-by-side. Here, hovering over a summary sentence emboldens that sentence and its corresponding aligned source text [5]. Additionally, clicking a summary sentence assigns a persistent blue background to the aligning texts [6]. Users can edit the summary freely, with alignments updating automatically.

interaction (HCI) methodologies and applying prominent usability questionnaires. These studies indicate the system's utility and user-friendly design for a thorough collaborative summarization process. Notably, users valued the tool's guidance throughout the process, while also appreciating their continuous involvement in refining automatic decisions.

2 Background and Related Work

This section provides a brief overview of related lines of work in summarization. These include strategies offering some level of user control (§2.1),

and modular summarization pipelines that separate the task into distinct subtasks (§2.2).

2.1 User Impact on the Summarization Process

Several previous lines of research focused on giving users control of the summary content. In tasks like query-focused (Dang, 2006; Baumel et al., 2018) and aspect-based summarization (Ahuja et al., 2022; Yang et al., 2023), the input text is accompanied by a request around which to focus the output summary. This is a common non-interactive approach for guiding summary content. Other works adapt the summarization process to specific users by learning their preferences. Hu et al. (2012) and Tepper et al. (2018) profile users in order to personalize the summary, via previously discussed aspects in conversations and social connections. Similarly, research on active learning collects summary preferences from users and learns their inclinations toward content and format in order to improve the model's performance (P.V.S and Meyer, 2017; Zarinbal et al., 2019; Gao et al., 2020). In these works, user influence is mainly confined to attributes in the input or during model adaptation, leaving the summarization process itself fully automatic. In contrast, our approach supports complete user control and intervention in both content selection and the post-generation phase.

Another line of work focuses on designing interactive tools that provide users with certain means of intervention *during* the summarization process. Yan et al. (2011) developed a system supporting iterative selection and removal of source sentences in an extractive system summary until a satisfactory summary is obtained. To aid users in making informed decisions, the system helps users track the context in which summary sentences were mentioned in the source texts. Similarly, P.V.S. et al. (2018) introduced a tool where users can iteratively select concepts in a system summary to remove from the summary or upon which to further elaborate. Xie et al. (2023)'s system allows users to edit system summaries by typing text and receiving automatic completion suggestions. Despite facilitating collaboration with users, these tools start with complete generic system summaries before integrating user feedback. Specifically, they are not well-suited for cases where users wish to include content not present in the initial system summary, or for completely changing its content. In contrast, our system adapts to user feedback throughout the entire process, allowing users to choose what to include in the summary and assisting them in editing the output to further adjust it to their preferences.

Lastly, interactive exploration systems (Shapira et al., 2022) provide updated summaries for given queries. However, unlike SUMMHELPER, such systems aim to allow learning about a topic, rather than generating a coherent fine-tuned summary.

2.2 Modular Summarization

SUMMHELPER is a modular system consisting of separate components, each performing one subtask, allowing user modifications of that sub-task's output. Such decomposition has been studied before in the context of fully automated summarization, with several works separating the process into salience detection and generation components (Barzilay and McKeown, 2005; Li et al., 2018; Ernst et al., 2022). These works focused on optimizing each component as part of a fully-automatic summarization process in order to improve the overall performance of the model. In contrast, our work uses this modularity to not only improve overall system output, but to also give more control to the user over each step in the summarization process.

3 The SUMMHELPER Application

SUMMHELPER is a web application designed for human-computer cooperation in generating human-controlled summaries, shown in Figure 1. It consists of two stages: (i) computer-assisted content selection via highlighting (§3.1), and (ii) automated summary generation according to the selected content followed by machine-assisted reviewing and editing of the generated summary (§3.2).

3.1 Personalized Content Selection

The first step focuses on content selection. The information to incorporate in the summary is manually selected by highlighting it via mouse click-and-drag operations ([3] in Figure 1a). Notably, users can also get suggested content from SUMMHELPER ([1], pale yellow), by clicking the magic wand icon ([2]). Users can accept or reject a full suggestion by clicking the ✓ and ✗ buttons, respectively, which appear when hovering over the suggestion ([4]).

To automatically identify suggested highlights, we deploy the ExtractiveSummarizer model from

the TransformerSum library.[2] The model, a RoBERTa$_{base}$ (Liu et al., 2019) trained on the CNN/DailyMail summarization dataset (Hermann et al., 2015), operates as a binary classifier. Its function is to assess the significance of each sentence within the text. As a subsequent operation, the application selects the 30% highest-ranking sentences to suggest to the user. The choice of this model was influenced by its popularity among extractive summarizers, which are all trained to predict salience. Yet, it can be easily replaced with other content selection models to cater to varying needs.

We note that these recommendations are primarily applicable for *generic* summaries. The final content selection decision lies with the users, whose judgment and scrutiny of these suggestions, along with the additional selection of non-suggested content, is instrumental in tailoring the summary to their specific preferences.

3.2 Content Consolidation

Once all the desirable content is selected, the next step is to properly consolidate it into a coherent summary. In our setting, SUMMHELPER initially auto-generates such a summary, subsequently providing users with guidance for its review and refinement. For the initial auto-consolidation, we deploy an available Controlled Text Reduction model (Slobodkin et al., 2023), which is a Flan-T5$_{large}$ model (Chung et al., 2022), finetuned on the highlights-focused CTR dataset.[3] Upon generation, users are presented with the generated summary and the highlighted input text side-by-side (see Figure 1b). This view facilitates reviewing the summary and editing it when identifying unfavorable outcomes, such as the absence of highlighted content or the inclusion of undesired (non-highlighted or hallucinated) content. To facilitate examination of the summary's compliance with the highlighted content, the user can hover over summary sentences to embolden both the summary sentence and its corresponding alignment in the source text ([5]). An alignment can be permanently emphasized with a blue background by clicking on a summary sentence, which remains unaffected when hovering over other sentences ([6]). To ensure consistent alignment while the summary is being revised by the user, SUMMHELPER monitors writing pauses and re-calculates alignments when a pause exceeds one second.

Considering the computational demands of continuous on-the-fly re-alignment, and the alignment feature's primary goal of pointing users to relevant source text sections, we opted for a lexical-matching approach, which is both fast and sufficient for this goal.[4] Our approach locates the longest common subsequence (LCS) between the lemmas of each input sentence and each summary sentence, followed by several heuristics to filter out irrelevant LCSs (see Appendix A for further details).

User	1	2	3	4	5	6
SUS Score	95	95	90	67.5	90	82.5

Table 1: SUS scores for each user, calculated based on the ten SUS question scores (see Appendix C.1).

4 Experiments and Evaluation

We assess SUMMHELPER via two user studies with human subjects, using standard human-computer interaction (HCI) questionnaires. In the first study, we examine the usability of SUMMHELPER for carrying out its purpose, i.e., summarizing an article in a collaborative manner, granting control to the user throughout the process. The second study compares SUMMHELPER to a conventional summarization setup, where a standard auto-generated summary can simply be post-edited without any specialized automated assistance, aiming to assess SUMMHELPER's comparative utility.

4.1 Usability Study

Setup. This study aims to gather human feedback regarding the usefulness of SUMMHELPER in performing a collaborative, user-guided, summarization process. Following the discount usability testing principle (Nielsen, 1993), which contends that six evaluators are sufficient for prototype evaluation, we employed six participants for this study. To simulate a plausible real-world scenario, participants were given the persona of an intern journalist who is required to use the application for writing a summary of a news article. All participants performed the task twice, over the same two articles, taken from the DUC 2001 dataset,[5] in random order.

[2] https://transformersum.readthedocs.io/en/latest/

[3] For further details, see Appendix B.

[4] Semantic matching was examined during system development, but was found to have little added value with substantially higher latency.

[5] https://duc.nist.gov

System Aspect	Score
Highlights suggestion model	3.7 (1.0)
Alignments algorithm	4.3 (1.0)
CTR model	
Summary coherence	4.2 (0.7)
Summary non-redundancy	4.6 (0.4)
Highlights coverage	4.7 (0.4)
Highlights adherence	4.2 (0.7)
Overall satisfaction	4.0 (0.7)
General	
Intuitiveness of highlighting	4.5 (0.4)
Likeliness to recommend	4.2 (0.7)

Table 2: The average and (StD) results of the Usefulness questionnaire on the 12 sessions (2 articles for 6 participants). See Appendix C.1 for the full questions.

To assess SUMMHELPER's helpfulness in different use cases, one article was relatively long, with ∼800 tokens, whereas the other contained ∼500 tokens.

During the experiments, we observed the users' activity and employed a "think aloud" technique (Van Someren et al., 1994) to obtain user remarks. Upon completing the summaries of both articles, participants filled out the standard System Usability Scale (SUS) questionnaire (Brooke, 1996) for subjective usability evaluation, consisting of questions regarding the system's ease of use, ease of learning, and general flow, with an overall score between 0 and 100.

Additionally, after summarizing each article, participants rated the usefulness of various characteristics of the application on a 1 to 5 scale, including the quality of the different models and algorithms used in the system, the intuitiveness of highlighting and unhighlighting content, and the likeliness of them recommending the system. For more details about the setup, including the full list of the SUS questions and our additional questions, see Appendix C.1.

Results. Table 1 presents the SUS scores of each of our 6 participants. With the exception of user number 4,[6] the system received scores exceeding 80, thereby affirming the application's "excellent" usability (UIUX-Trend, 2021). See Table 4 in the Appendix for itemized scores.

[6]This single participant expressed a strong personal preference for a more abstractive automatic summary, even though this is not necessarily a desired goal on its own in our setting.

This favorable trend is further observed in Table 2, which outlines the average ratings on the system features, across the 12 sessions. Overall, users expressed satisfaction with the application, finding SUMMHELPER's features helpful and intuitive, including the initial highlight suggestions and the alignment feature. Furthermore, the generated summaries by the CTR model were viewed as highly satisfactory, and there was a discernible interest among several participants to incorporate such an application into their everyday work (e.g., for summarizing legal contracts as well as prescription drug information).

During the study, we observed that the majority of users felt that the suggested highlights were particularly useful when navigating through the *longer* article as opposed to the *shorter* one. Nevertheless, all users expressed satisfaction with the overall summarization process of SUMMHELPER for both articles. They particularly appreciated the two-step procedure encompassing content selection and subsequent review, as it facilitated better text comprehension and instilled greater confidence and control in producing the final output. Two users expressed a desire for an option to create more abstractive summaries that are less verbatim relative to the highlights. Addressing this feedback, by training more abstractive CTR models or performing a post-hoc abstraction of the generated summary, is an interesting future direction we plan to explore. See Appendix C.1 for more feedback and issues raised by participants.

4.2 Comparative Usability Test

Setup. We compared the use of SUMMHELPER with a setup that simulates a conventional approach when working with summarization systems. In such setup, the input text is first generically summarized with an automatic summarization model. That summary can then be manually post-edited to meet the specific preferences of the user. For the summarization model, we used a $\text{BART}_{\text{large}}$ model (Lewis et al., 2019) trained on the CNN/Daily Mail dataset (Hermann et al., 2015),[7] selected for its noticeable popularity. We adapted SUMMHELPER's front-end for this process in order to eliminate a potential influence caused by the application's design. The resulting application comprises two steps: the automatic generation of the generic summary

[7]https://huggingface.co/facebook/bart-large-cnn

Dimension	Score
Usefulness	4.3 (0.5)
Ease of Use	3.6 (0.6)
Ease of Learning	3.1 (0.3)
Satisfaction	4.1 (0.6)
Summarization Process	4.7 (0.5)

Table 3: The average (StD) results of the five dimensions in the USE questionnaire. A score of 1 represents a preference for ONLYSUMM and 5 prefers SUMMHELPER.

and the review step for manual editing. During reviewing, users are presented with the input text and the generated summary side-by-side, allowing them to make adaptations to the summary (without the alignment feature). We refer to this adapted application as ONLYSUMM.

For this experiment, we asked 6 new participants to follow the task described in §4.1, which involved summarizing a news article, taking the perspective of an intern journalist, once with SUMMHELPER on one article, and once with ONLYSUMM on another article (with different orders of articles and applications). Upon completion of both sessions, participants filled out a questionnaire, adapted from the standard USE Questionnaire (Lund, 2001). In the questionnaire, 32 statements are rated on a scale of 1 (ONLYSUMM is preferred) to 5 (SUMMHELPER is preferred). The original 30 USE statements represent 4 dimensions: Usefulness, Ease of Use, Ease of Learning, and Satisfaction (see Appendix C.2 for the full list of statements). We also added 2 statements to rank users' experience with the key aspects of our summarization process (represented as the fifth dimension in Table 3). These additional statements were: "I found it easy to control what information to include in the final summary" and "I found it easy to make sure the final summary had all the information I wanted". More details elaborating on the study are available in Appendix C.2.

Results. Table 3 presents the scores for each dimension examined, averaged over the corresponding statements and the six participants. Interestingly, despite SUMMHELPER consisting of more features and steps than ONLYSUMM, participants did not find it more challenging to learn. Moreover, they reported that SUMMHELPER was somewhat more user-friendly. SUMMHELPER was strongly favored over ONLYSUMM in terms of Usefulness, Satisfaction, and, notably, the Summarization Process, underscoring the practicality of SUMMHELPER for preparing customized summaries.

Importantly, we observed that users tended to be very meticulous when summarizing with SUMMHELPER, exhibiting a higher inclination to carefully inspect the text and critically evaluate the inclusion of each piece of information. Indeed, even with the suggested highlights, users cautiously appraised each suggestion and more often selected only sub-segments of it. In contrast, we found that when summarizing with ONLYSUMM, participants typically skimmed the input text and accepted the generated summaries with minimal adjustments. Therefore, although using SUMMHELPER generally took longer to summarize (11.1 minutes on average, compared to 7.0 minutes with ONLYSUMM), it led to a more thorough summarization process. This is corroborated by the Usefulness, Satisfaction, and Summarization Process scores in Table 3, and participants' feedback, which consistently indicated higher confidence and satisfaction with their completed work when using SUMMHELPER.

5 Conclusion

In this paper, we presented SUMMHELPER, a novel summarization assistant, which collaborates with users across two steps: content selection and content consolidation. The system facilitates user intervention and supervision along the summarization process, in order to achieve the most suitable output tailored to specific needs. Preliminary user studies illustrate SUMMHELPER's potential for a thorough and collaborative summarization process, with users expressing satisfaction with the process, as well as the final output.

Future work may include investigating more effective semantic strategies to locate summary-source alignments with acceptable latency. Additionally, in light of some user feedback, another interesting extension includes developing more abstractive consolidation and fusion models, which would offer control over the level of abstractness in the outputs. Lastly, exploring strategies to scale SUMMHELPER to a multi-document setting presents another promising avenue for future investigation.

Limitations

This demo focuses on the single-document setting. Future work should expand the application's capa-

bilities to the multi-document setting, both in terms of the backend models and in terms of accessibility and intuitiveness of the application's frontend design. Additionally, our tool currently helps users in the reviewing step solely with the alignment functionality. Future work should add additional assistance during this step in the form of suggested improvements to selected unsatisfactory content in the summary, in addition to the alignment feature.

Ethics Statement

We conducted the usability (§4.1) and comparative usability (§4.2) studies in person. Participants volunteered to take part in the study, taking about 40 minutes for the former experiment, or 35 minutes for the latter. A consent form was signed by participants prior to each session, which stressed the fact that the user study was voluntary and that they were encouraged to withdraw if they felt any discomfort. In addition, the form ensured that the participant is at least 18 years of age, and assured that personal details remain anonymous.

The source texts (news articles) used in the user studies were acquired according to the required NIST guidelines (`https://duc.nist.gov`).

Acknowledgements

This work was supported by the Israel Science Foundation (grant no. 2827/21), and a grant from the Israel Ministry of Science and Technology. We would also like to thank Hadar Ronen for her guidance in planning the user studies.

References

Ojas Ahuja, Jiacheng Xu, Akshay Gupta, Kevin Horecka, and Greg Durrett. 2022. ASPECTNEWS: Aspect-Oriented Summarization of News Documents. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6494–6506, Dublin, Ireland. Association for Computational Linguistics.

Regina Barzilay and Kathleen R. McKeown. 2005. Sentence Fusion for Multidocument News Summarization. *Computational Linguistics*, 31(3):297–328.

Tal Baumel, Matan Eyal, and Michael Elhadad. 2018. Query Focused Abstractive Summarization: Incorporating Query Relevance, Multi-Document Coverage, and Summary Length Constraints into seq2seq Models.

John Brooke. 1996. SUS: A Quick and Dirty Usability Scale. *Usability evaluation in industry*, 189(3):189–194.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling Instruction-Finetuned Language Models.

Hoa Trang Dang. 2006. DUC 2005: Evaluation of Question-Focused Summarization Systems. In *Proceedings of the Workshop on Task-Focused Summarization and Question Answering*, pages 48–55, Sydney, Australia. Association for Computational Linguistics.

Ori Ernst, Avi Caciularu, Ori Shapira, Ramakanth Pasunuru, Mohit Bansal, Jacob Goldberger, and Ido Dagan. 2022. Proposition-Level Clustering for Multi-Document Summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1765–1779, Seattle, United States. Association for Computational Linguistics.

Frank Flemisch, David A Abbink, Makoto Itoh, M-P Pacaux-Lemoine, and Gina Wessel. 2019. Joining the blunt and the pointy end of the spear: towards a common framework of joint action, human–machine cooperation, cooperative guidance and control, shared, traded and supervisory control. *Cognition, Technology & Work*, 21:555–568.

Yang Gao, Christian M. Meyer, and Iryna Gurevych. 2020. Preference-Based Interactive Multi-Document Summarisation. *Information Retrieval*, 23(6):555–585.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching Machines to Read and Comprehend.

Jean-Michel Hoc. 2000. From human–machine interaction to human–machine cooperation. *Ergonomics*, 43(7):833–843.

Po Hu, Donghong Ji, Chong Teng, and Yujing Guo. 2012. Context-Enhanced Personalized Social Summarization. In *Proceedings of COLING 2012*, pages 1223–1238, Mumbai, India. The COLING 2012 Organizing Committee.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. *CoRR*, abs/1910.13461.

Wei Li, Xinyan Xiao, Yajuan Lyu, and Yuanzhuo Wang. 2018. Improving Neural Abstractive Document Summarization with Explicit Information Selection Modeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1787–1796, Brussels, Belgium. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692.

Arnold M Lund. 2001. Measuring Usability with the USE Questionnaire. *Usability interface*, 8(2):3–6.

Jakob Nielsen. 1993. Usability Engineering.

Marie-Pierre Pacaux-Lemoine, Damien Trentesaux, Gabriel Zambrano Rey, and Patrick Millot. 2017. Designing intelligent manufacturing systems through Human-Machine Cooperation principles: A human-centered approach. *Computers & Industrial Engineering*, 111:581–595.

Avinesh P.V.S., Benjamin Hättasch, Orkan Özyurt, Carsten Binnig, and Christian M. Meyer. 2018. Sherlock: A System for Interactive Summarization of Large Text Collections. *Proc. VLDB Endow.*, 11(12):1902–1905.

Avinesh P.V.S and Christian M. Meyer. 2017. Joint Optimization of User-desired Content in Multi-document Summaries by Learning from User Feedback. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1353–1363, Vancouver, Canada. Association for Computational Linguistics.

Ori Shapira, Ramakanth Pasunuru, Mohit Bansal, Ido Dagan, and Yael Amsterdamer. 2022. Interactive Query-Assisted Summarization via Deep Reinforcement Learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2551–2568, Seattle, United States. Association for Computational Linguistics.

Aviv Slobodkin, Avi Caciularu, Eran Hirsch, and Ido Dagan. 2023. Dont add, dont miss: Effective content preserving generation from pre-selected text spans.

Aviv Slobodkin, Paul Roit, Eran Hirsch, Ori Ernst, and Ido Dagan. 2022. Controlled Text Reduction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5699–5715, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Naama Tepper, Anat Hashavit, Maya Barnea, Inbal Ronen, and Lior Leiba. 2018. Collabot: Personalized Group Chat Summarization. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, page 771–774, New York, NY, USA. Association for Computing Machinery.

UIUX-Trend. 2021. Measuring and Interpreting System Usability Scale - UIUX Trend. https://uiuxtrend.com/measuring-system-usability-scale-sus/. Accessed: 2023-08-01.

Maarten Van Someren, Yvonne F Barnard, and J Sandberg. 1994. The Think Aloud Method: A Practical Approach to Modelling Cognitive Processes. *London: AcademicPress*, 11:29–41.

Yujia Xie, Xun Wang, Si-Qing Chen, Wayne Xiong, and Pengcheng He. 2023. Interactive Editing for Text Summarization.

Rui Yan, Jian-Yun Nie, and Xiaoming Li. 2011. Summarize What You Are Interested In: An Optimization Framework for Interactive Personalized Summarization. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1342–1351, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Xianjun Yang, Kaiqiang Song, Sangwoo Cho, Xiaoyang Wang, Xiaoman Pan, Linda Petzold, and Dong Yu. 2023. OASum: Large-Scale Open Domain Aspect-based Summarization. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4381–4401, Toronto, Canada. Association for Computational Linguistics.

Marzieh Zarinbal, Azadeh Mohebi, Hesamoddin Mosalli, Razieh Haratinik, Zahra Jabalameli, and Farnoush Bayatmakou. 2019. A New Social Robot for Interactive Query-Based Summarization: Scientific Document Summarization. In *Interactive Collaborative Robotics*, pages 330–340, Cham. Springer International Publishing.

A Alignment Algorithm

In the reviewing phase, the system aids users in comparing the highlighted input text to the generated output summary, in order to spot any potential disapprovals in the summary. This is achieved by automatically identifying text from the input that aligns with each sentence in the summary, and clearly marking it (§3.2) for the user. To find these alignments, the system first performs sentence tokenization on the input source text and the generated summary. For each pair of summary and input sentences, it then calculates the longest common subsequence (LCS) of their lemmas.

To filter out insignificant alignments, LCSs containing less than three content tokens (neither stop words nor punctuation), denoted *short LCS*s, are disregarded. For instance, as demonstrated in Figure 2, the LCS "John eat today" between the first

sentences of the summary and input consists of three content words and is thus preserved. In contrast, the LCS "Mr. Smith" between the first summary sentence and the second input sentence, having only two content words, is discarded. For alignment *within highlights*, a short LCS is still retained if it covers at least 25% of the highlighted span's content lemmas. For instance, even though the LCS "he call me" of the last sentences of the summary and input in Figure 2 contains only one content lemma ("call"), it covers 100% of the highlight's content lemmas and is thus retained.

Finally, the alignment algorithm also addresses cases where the CTR model reorders content within input sentences. An LCS procedure is not well-suited for such situations. To this end, the algorithm iteratively calculates four LCSs for each pair of summary and input sentences. After each iteration, the part of the summary sentence contributing to the LCS is omitted, enabling shorter LCSs to be identified. For example, after identifying the LCS of the first sentences of the input and summary in Figure 2 ("John eat today"), the algorithm generates a variant of the summary sentence by excluding the LCS, resulting in "Mr. Smith said early". It then identifies the LCS "Mr. Smith" between this variant and the first input sentence, which is preserved as it covers 50% of the second highlighted span's content lemmas ("Mr.", "Smith", "tell", "mother").

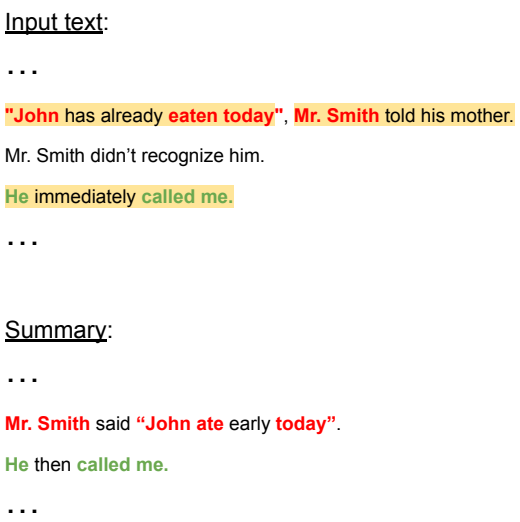


Figure 2: An example of the alignment algorithm for an extract of the highlighted input text and that of the respective summary. The first lemma-based LCS between the first sentences of the summary and input is *"John eat today"* (bold and red), which has ≥ 3 content words (John, eat, today) and is thus retained. The second LCS, *"Mr. Smith"*, contains $\geq 25\%$ of the second highlighted span's (*"Mr."*, *"Smith"*, *"tell"*, *"mother"*) content words, and is also retained. On the other hand, the LCS *"Mr. Smith"* between the first summary sentence and the second input sentence, having only two content words and lacking overlap with any highlighted span, is filtered out. For the second summary sentence and the third input sentence, the only LCS, *"He called me"* (bold and green), comprises a single content word (*"called"*) which covers 100% of the third highlight's content words and is thereby retained.

B CTR Model

Controlled Text Reduction (CTR; Slobodkin et al., 2022), is a recently introduced task, which takes as input a text with pre-selected marked spans ("highlights") and expects a coherent version of the text, covering exactly the content of these highlights. It handles coherence issues relating to discourse and coreference. This task conforms with our summary generation process, and we hence employ an available Controlled Text Reduction model.[8] This model is a a Flan-T5$_{\text{large}}$ model (Chung et al., 2022), finetuned on the highlights-focused CTR dataset. Following Slobodkin et al. (2022), highlights are incorporated into the input text with special markups, <*extra_id_1*> and <*extra_id_2*>, marking the beginning and end of each highlighted span, respectively. In our configuration, we set the maximum input length to 4096 and the maximum target length to 400. A greedy decoding strategy was used in order to optimize the decoding speed. Other parameters are kept consistent with the predefined generation parameters of the model.

[8] https://github.com/lovodkin93/CTR_instruction_finetuning

C Experimental Details

In this work, we performed a usability study and a system comparison experiment (§4) to assess the utility of our application.

C.1 System Usability Tests

For the usability study, six participants were gathered based on previous acquaintance. These participants varied in their age (28-33), gender, and occupation. Each session took approximately 40 minutes. A participant started by filling out an experiment participation consent form. Next, the different elements of the application were explained and demonstrated to the participant. Then, the participant was asked to experiment with the appli-

```
As an intern reporter, your assignment is to
study two articles written by a senior
journalist, and write a summary for each
article, suitable for sharing on social media
platforms.
This task forms a critical part of your
internship evaluation, hence meticulous
attention to detail is mandatory.
You're granted access to an application that
can assist you in accomplishing this task.
However, it's crucial that the final summary
remains a testament to your individual effort
and understanding of the articles.
```

Figure 3: The instructions given to the user study participants.

cation on an example article, to reduce the learning curve of using the system for the first time. Once this onboarding stage was over, the experimentee was presented with the assignment (see Figure 3). The participants conducted the experiments on two articles, one with ∼800 tokens and another with ∼500 tokens, in a random order.

SUS questionnaire. The System Usability Scale (SUS) questionnaire (Brooke, 1996) was filled out once by each participant after completing both article summaries, with the following 10 questions being rated on a scale from 1 ("strongly disagree") to 5 ("strongly agree"):

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

SUS Question	Average Score
I think that I would like to use this system frequently.	4.17 (0.98)
I found the system unnecessarily complex.	1.83 (0.98)
I thought the system was easy to use.	4.33 (0.52)
I think that I would need the support of a technical person to be able to use this system.	1.33 (0.82)
I found the various functions in this system were well integrated.	4.17 (0.41)
I thought there was too much inconsistency in this system.	1.17 (0.41)
I would imagine that most people would learn to use this system very quickly.	4.67 (0.52)
I found the system very cumbersome to use.	1.33 (0.52)
I felt very confident using the system.	4.50 (0.84)
I needed to learn a lot of things before I could get going with this system.	1.50 (0.84)

Table 4: The average (StD) score of the ten SUS questions asked after the usability study, on a scale of 1 to 5.

The SUS scores (Table 1) were calculated using the procedure by Brooke (1996), as follows. Initially, the score contributions from each item were summed up, with each item's contribution ranging in a 0 to 4 scale. For the odd-numbered items (1,3,5,7, and 9), the score contribution was determined as the scale position minus 1. Conversely, for the even-numbered items (2,4,6,8, and 10), the contribution was calculated as 5 minus the scale position. This sum was then multiplied by 2.5 to compute the overall SUS value for each user, with scores having a range of 0 to 100. We also calculated the average (StD) score for each question, as delineated in Table 4.

Usefulness questionnaire. After summarizing each of the two articles, the users filled out a usefulness questionnaire (see results in Table 2), where they were asked to rate the following 9 questions on a scale of 1 ("strongly disagree") to 5 ("strongly agree"), addressing the different components in our system:

1. For the requirements of the given task, the initial highlights were very helpful.
2. The alignments were helpful in assessing the content of the final summary.
3. It was intuitive to highlight and unhighlight information.
4. I would recommend this app for another intern journalist in my company.

Overall, the summary output by the system was:

5. Coherent
6. Non-Redundant
7. Highlights were covered fully
8. Did not cover unhighlighted content
9. To my satisfaction

Comments raised by participants. During the sessions, we collected comments and ideas for improvements raised by the participants. All the participants were very impressed with the summaries generated by the CTR model. Additionally, several users expressed their satisfaction with the modular process, stating that their continuous involvement was crucial for achieving the optimal summary. Users especially appreciated the side-by-side presentation of the highlighted input text and the summary, combined with the alignment feature, which helped them to both stay connected to the source text and optimize their navigation through it. For improvements, one suggestion was to enable generation of more abstractive summaries, that do not align as much with the highlights' phrasing. Additional suggestions included making a different icon for exiting erase mode and entering highlight mode in the content selection window,[9] enabling a dynamic number of suggestions proportionate to the text's length,[10] and enabling the option to go back to the beginning of the process by clicking the application's name in the toolbar.

C.2 System Comparison Experiment

For the comparative experiment, we gathered 6 new participants, also based on previous acquaintance. These participants varied in their age (24-35), gender, and occupation. Each session took approximately 35 minutes, which started with a participant filling out the same participation form as in the system usability tests (see Appendix C.1). Similarly to the usability test setting, prior to the actual experiment, the different elements of each of the two applications were explained and demonstrated to the participant, and they were asked to experiment with the system on an article. Once the participant felt confident with their understanding of each application, they were presented with the assignment in Figure 3 and asked to complete it on the same 2 articles as in the system usability tests, once with SUMMHELPER and once with ONLYSUMM (in different orders and different article-model pairings).

Questionnaire. After completing both articles, the participant answered a comparative usability questionnaire, adapted from the standard USE Questionnaire (Lund, 2001), as mentioned in §4.2. The original questionnaire consists of 30 statements, divided into 4 dimensions: Usefulness, Ease of Use, Ease of Learning, and Satisfaction. These questions are:

- Usefulness
  1. It helps me be more effective.
  2. It helps me be more productive.
  3. It is useful.
  4. It gives me more control over output.
  5. It makes it easier to achieve the desired output.
  6. It saves me time when I use it.
  7. It meets my needs in addressing the task.
  8. It does everything I would expect it to do.
- Ease of Use
  9. It is easy to use.
  10. It is simple to use.
  11. It is user-friendly.
  12. It requires the fewest steps possible to accomplish the task.
  13. It is flexible.
  14. Using it is effortless.
  15. I can use it without written instructions.
  16. I don't notice any inconsistencies as I use it.
  17. Both occasional and regular users would like it.
  18. I can recover from mistakes quickly and easily.
  19. I can use it successfully every time.
- Ease of Learning
  20. I learned to use it quickly.
  21. I easily remember how to use it.
  22. It is easy to learn to use it.

[9] In the first system version, there was only an icon to enter erasing mode, and in order to exit the erasing mode and enter highlighting mode, users needed to click this icon again.

[10] In the first system version, there were always 3 suggestions.

23. I quickly became skillful with it.

- Satisfaction

24. I am satisfied with it.
25. I would recommend it to a friend.
26. It is fun to use.
27. It works the way I want it to work.
28. It is wonderful.
29. I feel I need to have it.
30. It is pleasant to use.

For each statement, participants were asked to rate it on a scale from 1 (preferred ONLYSUMM) to 5 (preferred SUMMHELPER). In addition to those statements, we added two more statements, in order to rate the participants' experience with the key aspects of the Summarization Process:

31. I found it easy to control what information to include in the final summary.

32. I found it easy to make sure the final summary had all the information I wanted.

Observations and general feedback. Overall, all participants favored SUMMHELPER over ONLYSUMM. They especially appreciated the alignment feature, with one participant who started with SUMMHELPER, and expressed frustration with the absence of the alignment feature in ONLYSUMM. Additionally, we observed that all 6 users were meticulous when working with SUMMHELPER, and appraised each suggestion very carefully, as well as non-suggested content. Alternatively, when working with ONLYSUMM, 4 out of the 6 participants simply skimmed the article and were quick to accept the generated summary with minimal adjustments. This shows SUMMHELPER's potential to foster a more thorough and productive summarization process.