MDS*Writer*: Annotation tool for creating high-quality multi-document summarization corpora

Christian M. Meyer,[†‡] Darina Benikova,[†‡] Margot Mieskes,[†¶] and Iryna Gurevych[†‡]

[†] Research Training Group AIPHES
[‡] Ubiquitous Knowledge Processing (UKP) Lab,
Technische Universität Darmstadt, Germany
[¶] University of Applied Sciences, Darmstadt, Germany
http://www.aiphes.tu-darmstadt.de

Abstract

In this paper, we present MDS*Writer*, a novel open-source annotation tool for creating multi-document summarization corpora. A major innovation of our tool is that we divide the complex summarization task into multiple steps which enables us to efficiently guide the annotators, to store all their intermediate results, and to record user–system interaction data. This allows for evaluating the individual components of a complex summarization system and learning from the human writing process. MDS*Writer* is highly flexible and can be adapted to various other tasks.

1 Introduction

Motivation. The need for automatic summarization systems has been rapidly increasing since the amount of textual information in the web and at large data centers became intractable for human readers. While single-document summarization systems can merely compress the information of *one* given text, multi-document summaries are even more important, because they can reduce the actual *number* of documents that require attention by a human. In fact, they enable users to acquire the most salient information about a topic without having to deal with the redundancy typically contained in a set of documents. Given that most search engine users only access the documents linked on the first result pages (cf. Jansen and Pooch, 2001), multi-document summaries even have the potential to radically influence our information access strategies to such textual data that remains unseen by most current search practices.

At the same time, automatic summarization is one of the most challenging natural language processing tasks. Successful approaches need to perform several subtasks in a complex setup, including content selection, redundancy removal, and coherent writing. Training and evaluating such systems is extremely difficult and requires high-quality reference corpora covering each subtask.

Currently available corpora are, however, still severely limited in terms of domains, genres, and languages covered. Most of them are additionally focused on the results of only one subtask, most often the final summaries, which prevents the training and evaluation of intermediate steps (e.g., redundancy detection). A major corpus creation issue is the lack of tool support for complex annotation setups. Existing annotation tools do not meet our demands, as they are limited to creating final summaries without storing intermediate results and user interactions or are not freely available or support only single document summarization.

Contribution. In this paper, we present MDS*Writer*, a software for manually creating multi-document summarization corpora. The core innovation of our tool with regard to previous work is that we allow dividing the complex summarization task into multiple steps. This has two major advantages: (i) By linking each step to detailed annotation guidelines, we support the human annotators in creating high-quality summarization corpora in an efficient and reproducible way. (ii) We separately store the results of each intermediate step. This is necessary to properly evaluate the individual components of a complex automatic summarization system, but was largely neglected previously. Storing the intermediate results enables us to improve the evaluation setup beyond measuring inter-annotator agreement for the content selection and ROUGE for the final summaries.

Furthermore, we put a particular focus on recording the interactions between the users and the annotation tool. Our goal is to learn summa-

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics—System Demonstrations, pages 97–102,
Berlin, Germany, August 7-12, 2016.

rization writing strategies from the recorded user–system interactions and the intermediate results of the individual steps. Thus, we envision next-generation summarization systems that learn the human summarization *process* rather than trying to only replicate its *result*.

To the best of our knowledge, MDS*Writer* is the first attempt to support the complex annotation task of creating multi-document summaries with flexible and reusable software providing access to process data and intermediate results. We designed an initial, multi-step workflow implemented in MDS*Writer*. However, our tool is flexible to deviate from this initial setup allowing a wide range of summary creation workflows, including single-document summarization, and even other complex annotation tasks. We make MDS*Writer* available as open-source software, including our exemplary annotation guidelines and a video tutorial.[1]

2 Related work

There is a vast number of general-purpose tools for annotating corpora, for example, WebAnno (Yimam et al., 2013), Anafora (Chen and Styler, 2013), CSNIPER (Eckart de Castilho et al., 2012), and the UAM CorpusTool (O'Donnell, 2008). However, neither of these tools is suitable for tasks that require access to multiple documents at the same time, as they are focused on annotating linguistic phenomena within single documents or search results with limited contexts.

Tools for cross-document annotation tasks are so far limited to event and entity co-reference, e.g., CROMER (Girardi et al., 2014). These tools are, however, not directly applicable to the task of multi-document summarization. In fact, all tools discussed so far lack a definition of complex annotation workflows spanning multiple steps, which we consider necessary for obtaining intermediate results and systematically guiding the annotators.

With regard to the user–system interactions, the work on the Webis text reuse corpus (Potthast et al., 2013) is similar to ours. They ask crowdsource workers to retrieve sources for a given topic and record their search and text reuse actions. However, they approach a plagiarism detection task and therefore focus on writing essays rather than summaries and they do not provide detailed guidelines which is necessary to create high-quality corpora.

Summarization-specific software tools address the assessment of written summaries, computer-assisted summarization, or the manual construction of summarization corpora. The Pyramid annotation tool (Nenkova and Passonneau, 2004) and the tool[2] used for the MultiLing shared tasks (Giannakopoulos et al., 2015) are limited to comparing and scoring summaries, but do not provide any writing functionality. Orăsan et al.'s (2003) CAST tool assists users with summarizing a document based on the output of an automatic summarization algorithm. However, their tool is restricted to single-document summarization.

The works by Ulrich et al. (2008) and Nakano et al. (2010) are most closely related to ours, since they discuss the creation of multi-document summarization corpora. Unfortunately, their proposed annotation tools are not available as open-source software and thus cannot be reused. In addition to that, they do not record user–system interactions, which we consider important for next-generation automatic summarization methods.

3 MDS*Writer*

MDS*Writer* is a web-based tool implemented in Java/JSP and JavaScript. The user interface consists of a dashboard providing access to all annotation projects and their steps. Each step communicates with a server application that is responsible for recording the user–system interactions and the intermediate results. Below, we describe our proposed setup with seven subsequent steps motivated by our initial annotation guidelines.

Dashboard. Our tool supports multiple users and topics (i.e., the document sets that are to be summarized). After logging in, a user receives a list of all topics assigned to her or him, the number of documents per topic, and the status of the summarization process. That is, for each annotation step, the tool shows either a green checkmark (if completed), a red cross (if not started), or a yellow circle (if this step comes next) to indicate the user's progress. By clicking on the yellow circle of a topic, the user can continue his or her work. Figure 2 (a) shows an example with ten topics.

Step 1: Nugget identification. The first step aims at the selection of salient information within the multiple source documents, which is the most important and most time-consuming annotation step. Figure 1 shows the overall setup. Two thirds

[1] https://github.com/UKPLab/mdswriter

[2] http://143.233.226.97:60091/MMSEvaluator/

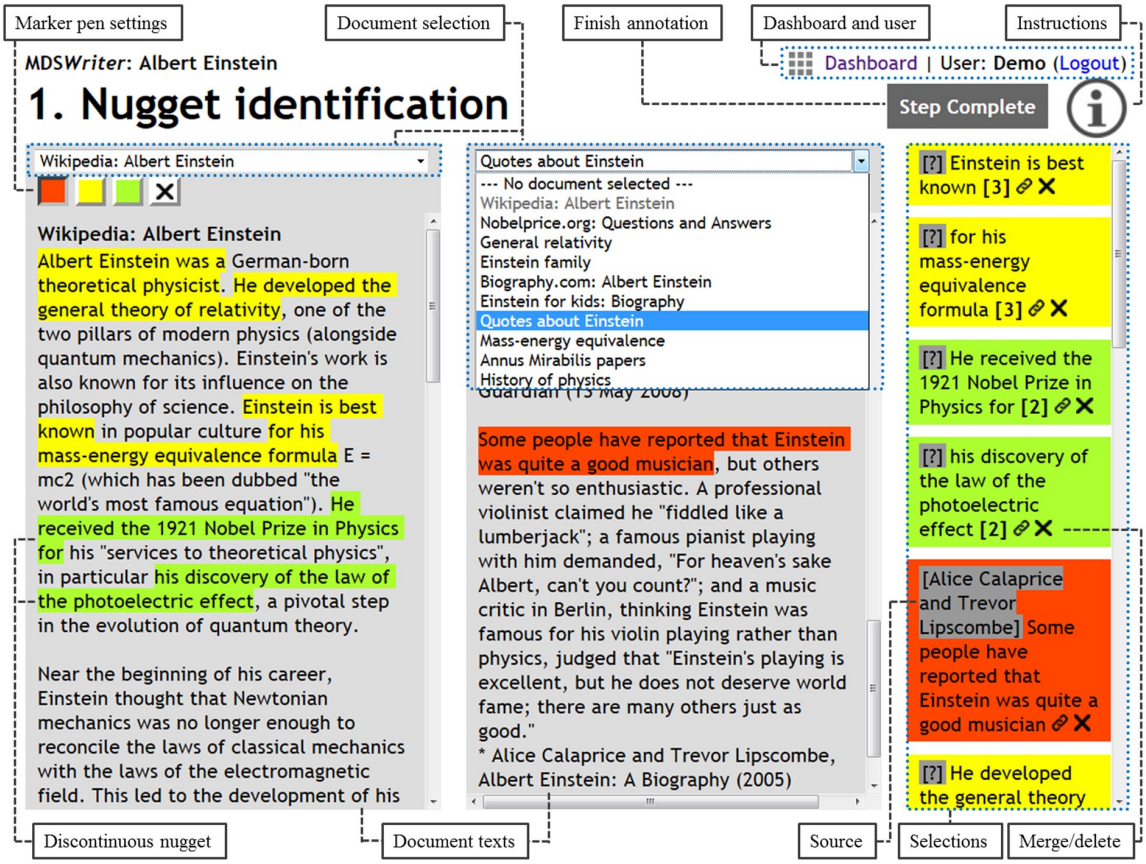

Figure 1: Nugget identification step with explanations of the most important components

of the screen are reserved for displaying the source documents. In the current setup, a user can choose to view a single source document over the full width or display two different source documents next to each other. The latter is useful for comparison and for ensuring consistent annotation.

Analogous to a marker pen, users can select salient parts of the text with the mouse. When releasing the mouse button, the selected text part receives a different background color and the selected text is included in the list of all selections shown on the right-hand side. The users may use three different colors to organize their selections. Existing selections can be modified and deleted.

To systematize the selection process, we define the term *important information nugget* in our annotation guidelines. Each nugget should consist of at least one verb with at least one of its arguments. It should be important, topic-related, and coherent, but not cross sentence boundaries. Typically, each selection in the source document corresponds to one nugget. But nuggets might also be discontinuous in order to support the exclusion of parenthetical phrases and other information of minor importance. Our tool models these cases by defining two distinct selections and merging them by means of a dedicated merge button.

Two special cases are nuggets referring to a certain source (e.g., a book) and nuggets within direct or indirect speech, which indicate a speaker's opinion. In figure 1, the 2005 biography is the source for the music-related selection. If a user would select only the subordinate clause, *some people* would be the speaker. As information about the source or speaker is highly important for both automatic methods and human writers, we provide a method to select this information within the text. The selection list shows the source/speaker in gray color at the beginning of a selection (default: [?]).

Having finished the nugget identification for all source documents, a user can return to the dashboard by clicking on "step complete".

Step 2: Redundancy detection. Redundancy is a key characteristic of multiple documents about a given topic. Automatic summarization methods aim at removing this redundancy. But, at the same time, most methods rely on the redundancy signal when estimating the importance of a phrase or sentence. Therefore, our annotation guidelines for

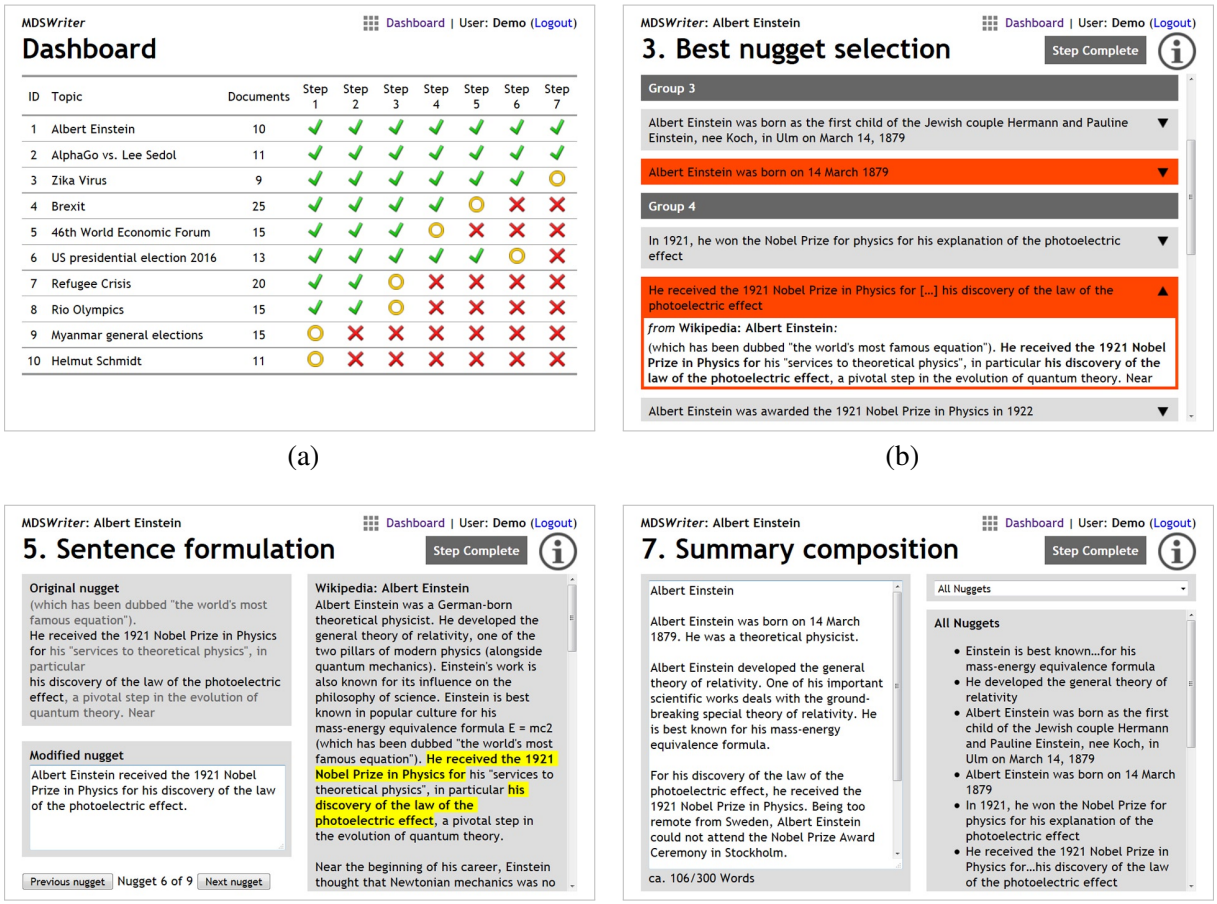

Figure 2: Screenshots of the dashboard (a) and the steps 3 (b), 5 (c), and 7 (d)

step 1 suggest to identify *all* important nuggets, including redundant ones. This type of intermediate result will allow us to create a better setup for evaluating content selection algorithms than comparing their outcome to redundancy-free summaries.

As our ultimate goal is, however, to compose an actual summary, we still need to remove the redundancy, which motivates our second annotation step. Each user receives a list of his or her extracted information nuggets and may now reorder them using drag and drop. As a result, nuggets with the same or a highly similar content will yield a single group. To allow for an informed decision, users may expand each nugget to view a context box showing the title of the source document and a ± 10 words window around the nugget text.

Step 3: Best nugget selection. In the third step, users select a representative nugget from each group, which we call the *best nugget*. We guide their decision by suggesting to prefer declarative and objective statements and to minimize context dependence (e.g., by avoiding deixis or anaphora). To select the best nugget, users can click on one of the nuggets within a group, which then turns red. Users may change their decisions and open a context box similar to step 2. Figure 2 (b) shows an example with two groups and a context box.

Step 4: Co-reference resolution. Although the users should avoid nuggets with co-references in step 3, there is often no other choice. Therefore, we aim at resolving the remaining co-references as part of a fourth annotation step. Even though human writers make vast use of co-references in a final summary, they usually change them with regard to the source documents. For example, it is uncommon to use a personal pronoun in the very first sentence of a summary, even if this cataphor would be resolved in the following sentences. Therefore, our approach is to first resolve *all* co-references in the best nuggets during step 4 and establish a meaningful discourse structure later when composing the actual summary in step 7.

To achieve this, MDS*Writer* displays one best nugget at a time and allows the user to navigate

through them. For each best nugget, we show its direct context, but also provide the entire source document in case the referring expression is not included in the surrounding ten words.

Step 5: Sentence formulation. Since our notion of information nuggets is on sub-sentence level, we ask our users to formulate each best nugget as a complete, grammatical sentence. This type of data will be useful for evaluating sentence compression algorithms, which start with an entire sentence extracted from one of the source documents and aim at compressing it to the most salient information. In our guidelines, we suggest that the changes to the nugget text should be minimal and that both the statement's source (step 1) and the resolved co-references (step 4) should be part of the reformulated sentence. We use the same user interface as in the previous step. That is, we display a single best nugget in its context and ask for the reformulated version. Figure 2 (c) shows a screenshot of a reformulated discontinuous nugget.

Step 6: Summary organization. While important nuggets often keep their original order in single-document summaries, there is no obvious predefined order for multi-document summaries. Therefore, we provide a user interface for organizing the sentences (step 5) in a meaningful way to formulate a coherent summary. A user receives a list of her or his sentences and may change the order using drag and drop. Additionally, it is possible to insert subheadings (e.g., "conclusion").

We consider this step important as previous approaches, for example, by Nakano et al. (2010, p. 3127) "did not instruct summarizers about how to connect parts" and thus do not control for coherence. By explicitly defining the order, we get in a position to learn from the human summarization process and improve the coherence of automatically generated extracts.

The user interface for step 6 is similar to the steps 2 and 3. It shows a sentence list and allows opening a context box with the original nugget.

Step 7: Summary composition. Our final step aims at formulating a coherent summary based on the structure defined in step 6. MDS*Writer* provides a text area that is initialized with the reformulated (step 5) and ordered (step 6) best nuggets, which can be arbitrarily changed. In our setup, we ask the users to make only minimal changes, such as introducing anaphors, discourse connectives, and conjunctions. This will yield summaries that are very close to the source documents, which is especially useful for evaluating extractive summarization methods. However, MDS*Writer* is not limited to this procedure and future uses may strive for abstractive summaries that require substantial revisions.

While writing the summary, the users have access to all source documents, to their original nuggets (step 1) and to their selection of best nuggets (step 3). By means of a word counter, the users can easily produce summaries with a certain word limit. Figure 2 (d) shows the corresponding user interface. Having finished their summary, users complete the entire annotation process for the current topic and return to the dashboard.

Server application. Each user action, ranging from the selection of a new nugget (step 1) to modifications of the final summary (step 7), is automatically sent to our server application. We use a WebSocket connection to ensure efficient bidirectional communication. The user–system interactions and all intermediate results are stored in an SQL database. Conversely, the server loads previously stored inputs, such that the users can interrupt their work at any time without losing data.

The client–server communication is based on a simple text-based protocol. Each message consists of a four character operation code (e.g., `7DNE` indicating that step 7 is now complete) and an arbitrary number of tab-separated parameters. The message `1NGN 1 25 100 2` indicates, for example, that the current user added a new nugget of length 100 characters to document 1 at offset 25, which will be displayed in color 2 (yellow).

4 Extensibility

The annotation workflow discussed so far is one example of dividing the complex setup of multi-document summarization into clear-cut steps. We argue that this division is important to ensure consistent and reliable annotations and to record intermediate results and process data. Despite this exemplary setup, MDS*Writer* provides an ideal basis for many other summarization workflows, such as creating structured or aspect-oriented summaries. This can be achieved by rearranging already existing steps and/or adding new steps. To this end, we designed the bidirectional and easy-to-extend message protocol described in the previous section as well as a brief developer guide on GitHub.

Of particular interest is that MDS*Writer* features cross-document annotations, the recording of user–system interactions and intermediate results, which is also highly relevant beyond the summarization scenario. Therefore, we consider MDS*Writer* as an ideal starting point for a wide-range of other complex multi-step annotation tasks, including but not limited to information extraction (combined entity, event, and relation identification), terminology mining (selection of candidates, filtering, describing, and organizing them), and cross-document discourse structure annotation.

5 Conclusion and future work

We introduced MDS*Writer*, a tool for constructing multi-document summaries. Our software fills an important gap as high-quality summarization corpora are urgently needed to train and evaluate automatic summarization systems. Previously available tools are not well-suited for this task, as they do not support cross-document annotations, the modeling of complex tasks with a number of distinct steps, and reusing the tools under free licenses. As a key property of our tool, we store all intermediate annotation results and record the user–system interaction data. We argued that this enables next-generation summarization methods by learning from human summarization strategies and evaluating individual components of a system.

In future work, we plan to create and evaluate an actual corpus for multi-document summarization using our tool. We also plan to provide monitoring components in MDS*Writer*, such as computing inter-annotator agreement in real-time.

Acknowledgements. This work has been supported by the DFG-funded research training group "Adaptive Preparation of Information form Heterogeneous Sources" (AIPHES, GRK 1994/1) and by the Lichtenberg-Professorship Program of the Volkswagen Foundation under grant № I/82806.

References

Wei-Te Chen and Will Styler. 2013. Anafora: A Web-based General Purpose Annotation Tool. In *Proceedings of the 2013 NAACL/HLT Demonstration Session*, pages 14–19, Atlanta, GA, USA.

Richard Eckart de Castilho, Sabine Bartsch, and Iryna Gurevych. 2012. CSniper – Annotation-by-query for Non-canonical Constructions in Large Corpora. In *Proceedings of the 50th Annual Meeting of the ACL: System Demonstrations*, pages 85–90, Jeju Island, Korea.

George Giannakopoulos, Jeff Kubina, John Conroy, Josef Steinberger, Benoit Favre, Mijail Kabadjov, Udo Kruschwitz, and Massimo Poesio. 2015. MultiLing 2015: Multilingual Summarization of Single and Multi-Documents, On-line Fora, and Call-center Conversations. In *Proceedings of the 16th Annual Meeting of the SIGDIAL*, pages 270–274, Prague, Czech Republic.

Christian Girardi, Manuela Speranza, Rachele Sprugnoli, and Sara Tonelli. 2014. CROMER: a Tool for Cross-Document Event and Entity Coreference. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, pages 3204–3208, Reykjavik, Iceland.

Bernard J. Jansen and Udo Pooch. 2001. A review of Web searching studies and a framework for future research. *Journal of the American Society for Information Science and Technology*, 52(3):235–246.

Masahiro Nakano, Hideyuki Shibuki, Rintaro Miyazaki, Madoka Ishioroshi, Koichi Kaneko, and Tatsunori Mori. 2010. Construction of Text Summarization Corpus for the Credibility of Information on the Web. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation*, pages 3125–3131, Valletta, Malta.

Ani Nenkova and Rebecca Passonneau. 2004. Evaluating Content Selection in Summarization: The Pyramid Method. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL*, pages 145–152, Boston, MA, USA.

Mick O'Donnell. 2008. Demonstration of the UAM CorpusTool for Text and Image Annotation. In *Proceedings of the 46th Annual Meeting of the ACL: Demo Session*, pages 13–16, Columbus, OH, USA.

Constantin Orăsan, Ruslan Mitkov, and Laura Hasler. 2003. CAST: A computer-aided summarisation tool. In *Proceedings of the 10th Conference of the European Chapter of the ACL*, pages 135–138, Budapest, Hungary.

Martin Potthast, Matthias Hagen, Michael Völske, and Benno Stein. 2013. Crowdsourcing Interaction Logs to Understand Text Reuse from the Web. In *Proceedings of the 51st Annual Meeting of the ACL*, pages 1212–1221, Sofia, Bulgaria.

Jan Ulrich, Gabriel Murray, and Giuseppe Carenini. 2008. A Publicly Available Annotated Corpus for Supervised Email Summarization. In *Enhanced Messaging: Papers from the 2008 AAAI Workshop*, Technical Report WS-08-04, pages 77–82. Menlo Park, CA: AAAI Press.

Seid Muhie Yimam, Iryna Gurevych, Richard Eckart de Castilho, and Chris Biemann. 2013. WebAnno: A Flexible, Web-based and Visually Supported System for Distributed Annotations. In *Proceedings of the 51st Annual Meeting of the ACL: System Demonstrations*, pages 1–6, Sofia, Bulgaria.