Interactive Abstractive Summarization for Event News Tweets

Ori Shapira[1], Hadar Ronen[2], Meni Adler[1], Yael Amsterdamer[1], Judit Bar-Ilan[2] and Ido Dagan[1]

[1]Department of Computer Science, Bar-Ilan University,Israel
[2]Department of Information Science, Bar-Ilan University, Israel
{obspp18,hadarg,meni.adler}@gmail.com
{yael.amsterdamer,judit.bar-ilan}@biu.ac.il, dagan@cs.biu.ac.il

Abstract

We present a novel interactive summarization system that is based on *abstractive* summarization, derived from a recent consolidated knowledge representation for multiple texts. We incorporate a couple of interaction mechanisms, providing a bullet-style summary while allowing to attain the most important information first and interactively drill down to more specific details. A usability study of our implementation, for event news tweets, suggests the utility of our approach for text exploration.

1 Introduction

Multi-document summarization (MDS) techniques aim to assist readers in obtaining the most important information when reading multiple texts on a topic. The dominant MDS approach focuses on constructing a short summary of some targeted length, capturing the most important information, mimicking a manually-crafted "static" summary. As an alternative, few papers considered *interactive summarization*, where the presented information can be interactively explored by the user according to needs and interest Christensen et al. (2014); Leuski et al. (2003); Yan et al. (2011).

In this paper we propose further contribution to this approach, focusing on interactive *abstractive* summarization. We suggest that an abstractive summarization approach, based on extracted "atomic" facts, is particularly suitable in the interactive setting as it allows more flexible information presentation. Intuitively, it makes more sense for a user to explore information at the level of individual facts, rather than the coarser level of full original sentences, as in prior work on interactive *extractive* summarization (see Section 6).

We build on the abstractive approach in supporting two useful modes of interaction. First, we present information in a *bullet-style summary*, where the most important information is initially displayed in bullet sentences, while further details may be obtained by *unfolding* additional bullets. Specifically, we implemented this approach for summarizing news tweets on a certain event along a time line (see Figure 2). Our second mode of interaction is *concept expansion*, which allows viewing complementary

information about a concept via its alternative term mentions, while tracking the concept occurrences throughout the summary (see Figure 3). This information is hidden in static summaries that use original sentences (extractive) or a single term per concept (abstractive).

To facilitate the modular construction of interactive summaries, we utilize as input a consolidated representation of texts, in particular the recent Open Knowledge Representation (OKR) of Wities et al. (2017). Briefly, this representation captures the propositions of the texts, where co-refferring concepts or propositions are collapsed together while keeping links to the original mentions (see Section 2). We leverage OKR structures to extract information at the level of atomic facts, to expand information from collapsed mentions and to retrieve the sources from which summary sentences were derived.

The novelties of our interactive scheme call for verifying its effectiveness and usefulness for users. For that, we have implemented our approach in a prototype system (Sections 3-4). This system automatically produces an interactive summary from input OKR data, which we assume to be parsed from original texts by an external black-box tool. We have examined our system through a set of standard usability tests Brooke (1996); Lund (2001) on gold standard OKR datasets that enabled us to study its contribution in isolation (Section 5). Our results show that the proposed system is highly valuable for readers, providing an appealing alternative to standard static summarization.

2 Preliminaries

As mentioned above, our interactive summarization system is based on a consolidated representation for the information in multiple texts. We next review some background on such representations and then describe the particular Open Knowledge Representation that we use.

2.1 Consolidated Representation

Motivated by summarization and text exploration, recent work considered the consolidation of textual information in various structures. As prominent examples, the studies of Liu et al. (2015) and Li et al. (2016) construct graph-based representations whose nodes are predicates or arguments thereof, extracted from the original text, and the predicate-argument relations are captured by edges. Identical or coreferring concepts are collapsed in a single node.

Rospocher et al. (2016) present a more supervised approach where concepts in the graph are linked to DBPedia[1] entries. This along with other metadata is used to detect coreferences and disambiguate concepts.

None of these works considers interactive summaries, and in particular none incorporates sufficient data for our modes of user interaction. We next briefly review the Open Knowledge Representation recently introduced by Wities et al. (2017), which is used by our system.

[1] `http://wiki.dbpedia.org/`

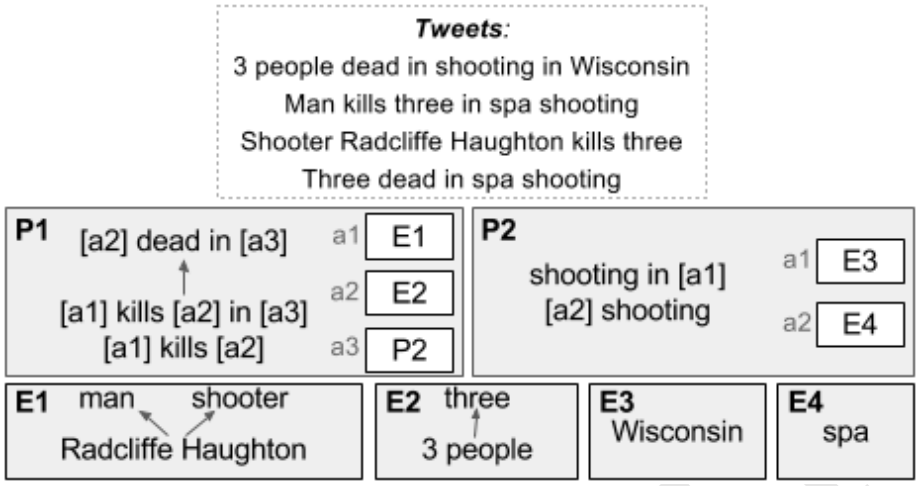

Figure 1: Four tweets on an event and their OKR structure.

2.2 Open Knowledge Representation

We illustrate the components of the OKR formalism that are central to our summarization method via the example OKR structure in Figure 1 (see Wities et al. (2017) for full details). On the top, there are four original tweets. On the bottom, there are two consolidated propositions (marked P1 and P2) and four entities (marked E1-E4) derived from these tweets. The figure depicts three types of links captured in OKR, as follows.

Mention links connect each proposition or entity with its set of mention terms, namely, every form of reference to the entity/proposition throughout the texts. E.g., E1 from Figure 1 is mentioned in the tweets as *"man"*, *"shooter"* or *"Radcliffe Haughton"*. Mentions of propositions are stored as *templates* with argument placeholders, e.g., *"[a2] dead in [a3]"*. Through their mentions, entities and propositions are further linked with their occurrences in the original texts (omitted from the figure).

Argument links connect propositions with their arguments, which may be entities or (nested) propositions. Since a proposition may have several templates with different arguments, argument IDs (marked a1-a3 in P1) are used to capture co-referring arguments within the same proposition. For example, a2 and a3 appear as arguments in the two templates of P1, and refer to entity E2 and proposition P2 respectively.

Entailment links, marked by directed edges in Figure 1, track semantic entailment (in context) between different types of OKR components. For example, in E1, *"Radcliffe Haughton"* entails *"man"* or *"shooter"*, namely, the former is more specific/informative in the *given context*.

3 Comprehensive Summary Information

The architecture of our system consists of two main steps: (1) a preprocessing step in which we generate comprehensive summary information and (2) interactive display of selected information. In this section we describe the first step, which is based on an input OKR structure. Our UI for exploring the summary information interactively is described in the following section.

The general scheme for generating summary information in our system is as follows.

1. Partition the OKR propositions into *groups*.
2. Generate representative *summary sentences* for each group of propositions. These yield the bullet-style summary sentences.
3. Generate metadata for each representative sentence: a *knowledge score*, *concept expansions* and *timestamp*.

For the current system, we implemented a baseline method for each of these steps, which nonetheless achieved high satisfaction scores in the usability study (see Section 5).

We partition propositions, as captured in the OKR structure, into distinct groups such that the propositions of each group are (transitively) connected by argument links (ignoring link direction). E.g., in Figure 1, P2 is nested in P1 and thus the two are grouped together.

Next, for the "root" (i.e., not nested) proposition in a group, we generate alternative candidate sentences. This is done by filling, in its templates, all the possible combinations of relevant argument mentions, and recursively so for nested propositions. For example, for P1 we would generate *"[3 people] dead in [shooting in [Wisconsin]]"*, *"[3 people] dead in [[spa] shooting]"*, *"[Three] dead in [[spa] shooting]"*, and so on (22 candidate sentences in total).

From each set of candidates we choose one representative sentence. Importantly, this means that unlike bounded-length summary paragraphs our comprehensive summary information effectively covers *all* the propositions in the original texts. Instead of filtering upfront less salient information, it is only hidden initially in the UI and can be unfolded by the user (see Section 4). For a representative sentence, we choose a candidate with high *language model score*,[2] high *knowledge score* (defined below) and small length. This is done by optimizing a weighted sum of these factors.

The knowledge score of each sentence intuitively reflects how *common* its mentions are in the original texts as well as how *informative* (specific) they are, based on the OKR entailment links. Reconsidering Figure 1 for example, in the tweet *"Three dead in spa shooting"*, the concepts *"three"*, *"dead"* and *"spa shooting"* should be rewarded for appearing each in two tweets, but *"three"* should be rewarded less than *"3 people"*, which is more informative.

We use the following heuristically formulated equation to calculate the score of each generated sentence s:

$$\text{score}(s) = \sum_{m \in \text{mentions}(s)} \alpha + \beta \cdot \text{depth}(m)$$

where $\text{mentions}(s)$ are the mentions of *predicates* and *entities* in the sentence. $\text{depth}(m)$ assigns a given mention m its depth in the relevant lexical entailment graph within the OKR. We have empirically set $\alpha = 1$, $\beta = 0.1$.

Each concept (entity or proposition) in the summary sentences is linked to its mentions and original texts using the OKR. The set of mentions is cleaned from duplicates (strings with small edit distance), yielding the *concept expansion* for sets with > 1 different mentions. This gives extra information about concepts that otherwise might have been missed. In Figure 3, for example, the *"suspected gunman"* is also identified

[2]For the language model, we trained an LSTM model (`https://github.com/yandex/faster-rnnlm`) on a collection of 100M tweets.

Figure 2: The initial view of a summary about a shooting in a Wisconsin spa covering 109 tweets. Ten generated sentences cover the most salient information throughout these tweets, and are ordered along the event timeline.

as *"Jamaican"*. For the tweet summarization scenario, we also compute the *timestamp* of each representative sentence as the time of the first tweet mentioning its root proposition.

4 Interactive User Interface

We now describe the *web application*[3] we implemented, designed for the interactive exploration of multiple tweets on a specific event. Our backend is implemented in Python 2.7 and runs on a CentOS server. The frontend is implemented with the AngularJS library. JSON is used for data interchange.

Figure 2 shows the initial screen summarizing a set of 109 tweets about the shooting in a Wisconsin spa from our running example. Bullet-style sentences (generated as explained in Section 3) are displayed along the event timeline, in descending order of their timestamps. As an indication of salience, to the right of each sentence, a pie chart shows the "percentage" of knowledge it covers according to its normalized knowledge score. The pie chart on the top shows the total knowledge "covered" by the currently visible sentences.

Initially, only sentences exceeding a certain information score threshold are displayed, as a concise bullet-style summary of the event. Other sentences are *folded* (e.g., between timestamps 01:07 and 22:55 in Figure 2). The user can then decide whether and which sentences to unfold, according to (a) time intervals of interest on the timeline; (b) the number of folded sentences, as indicated in the middle of the line; and (c) the amount of additional knowledge to be unfolded, which is highlighted on the top pie chart when hovering over folded sentences. By repeatedly unfolding sentences,

[3] http://u.cs.biu.ac.il/~shapiro1/okr/

Figure 3: The *concept expansion* pop-up consisting of mentions referring to the same person as "suspected gunman", revealing further information (e.g. "Jamaican").

Figure 4: The tweets pop-up shows a scrollable pane with the source tweets for a generated sentence.

the user can gradually discover the full timeline of the event with *all* consolidated data from the tweets.

Another mode of discovering information is via *concept expansion*: hovering over a highlighted concept (e.g., *"suspected gunman"*) opens a pop-up with different mentions of the same concept in the summary (Figure 3); clicking it further highlights all of its coreferences in the summary. Finally, the user can also click the Twitter icon to inspect the source tweets (Figure 4).

5 System Usability Tests

To assess and improve the value of our system, we have conducted two usability studies employing standard usability tests. The tests were performed on a dataset of human annotated OKR structures (of the form of Figure 1) released by Wities et al. (2017). We took their 6 largest clusters of event tweets, of about 100 tweets each. This gold-standard dataset enabled us to *study in isolation* the merits of our novel system. Given the positive results that we report below, we plan, in future work, to integrate and study our system in a fully automated pipeline.

SUS Question	*Avg. Score*
I think that I would like to use this system frequently.	3.83
I found the system unnecessarily complex.	2.33
I thought the system was easy to use.	3.33
I think that I would need the support of a technical person to be able to use this system.	2.17
I found the various functions in this system were well integrated.	3.83
I thought there was too much inconsistency in this system.	1.67
I would imagine that most people would learn to use this system very quickly.	3.5
I found the system very cumbersome to use.	1.33
I felt very confident using the system.	3.67
I needed to learn a lot of things before I could get going with this system.	2.17

Table 1: The ten SUS questions asked after the usability study and the average answer score on a scale of 1 to 5.

User	1	2	3	4	5	6
SUS Score	70	80	95	72.5	82.5	27.5

Table 2: SUS scores for each user, calculated based on the ten SUS question scores.

5.1 Preliminary Usability Study

A first usability study was conducted with two goals: to examine the usefulness of our ideas and to understand user needs.

Methodology. The evaluation phase of a prototype requires only a few evaluators, according to the "discount" usability testing principle Nielsen (1993). Thus, six students not familiar with our project were recruited as evaluators. We asked them to perform a series of predefined tasks on one of the six selected events. During the system usage we observed the users' activity and employed a "think aloud" technique to obtain user remarks. Each on-screen activity was captured using "Debut Video Capturing Software"[4]. After performing all tasks, users were asked to fill the SU Scale (SUS) questionnaire Brooke (1996) for subjective usability evaluation.

Results. Table 1 lists the average scores obtained for each of the ten SUS questions, on a scale of 1 to 5. Overall, users found the prototype easy to use and showed willingness to use it frequently.

[4] `http://www.nchsoftware.com/capture/`

The SUS questionnaire yields an important single number in $[0, 100]$ representing a composite measure of the overall usability of the system. This number is calculated based on the ten question scores. As seen in Table 2, except for one dissatisfied user[5], the system received high scores ranging from 70 to 95. The observation and verbal reports during the test yielded a list of requirements that helped improve our prototype.

5.2 Comparative Usability Test

After updating our system to incorporate improvements obtained from the preliminary study, we conducted another comparative study to examine the relative effectiveness of our system.

Methology. We have compared our system, here denoted by IAS (for Interactive Abstractive Summary), with two baseline approaches:

- Tweet: a list of all the original tweets in the event dataset.
- Static: the full ordered list of sentences generated by our system (Section 3), with no interactive features nor metadata (such as concept expansion, knowledge scores, etc.).

As mentioned earlier, we have used the gold-standard OKR structures for 6 of the events released by Wities et al. (2017). Six users were each presented with two events in each interface (IAS, Tweet, Static), where the assignment of event to interface and order of interfaces were different for each user. The users explored the information that describes each event in the assigned interface, and at the end were asked to complete the USE Questionnaire Lund (2001).

This questionnaire required users to rank each of the three interfaces on a scale from 1 to 3 according to 33 statements. The original 30 USE statements represent four dimensions: Usefulness, Satisfaction, Ease of Use, and Ease of Learning. We added three statements to rank user's experience of knowledge exploration.[6]

Results. Table 3 shows the average rank of each interface in each of the examined dimensions. While our system was naturally somewhat more complex to use than the baselines, which only require reading, it consistently received the highest ranks in the dimensions of Usefulness, Satisfaction and Knowledge Exploration. This indicates that interactivity indeed provides substantial value to the user, regardless of the summary sentences (as evident by the comparison to baseline Static).

The ranked USE statements also serve as an indication for the *quality* of our summary when compared to the other baselines. Standard summarization metrics are designed for static summaries[7] and are thus not expressly adequate for our interactive system due to its content being dynamic and user-manipulated. Having demonstrated here that interactive summaries are useful, designing and conducting dedicated quality tests for interactive summaries is a priority in our future work.

[5]This user had software quality assurance background and seemed to inspect for very minor software and user experience bugs, which we have later addressed.

[6]The three additional statements are: The system motivated me to actively explore more information; The system made me feel that I know the highlights of the event; The system helped me notice the important details of the event.

[7]The ROUGE and Pyramid methods are the common metrics to evaluate summaries.

Dimension	Tweet	Static	IAS
Usefulness	2.1	1.8	**2.3**
Knowledge Exploration	2.0	1.8	**2.6**
Satisfaction	2.0	1.7	**2.3**
Ease of Use	**2.5**	2.3	2.1
Ease of Learning	**2.7**	2.5	2.3

Table 3: USE questionnaire dimensions score comparison of the three system interfaces on a scale of 1 to 3.

6 Related Work

A vast body of work has been dedicated to the problem of multi-text summarization. We focus here on the rather few studies that enhance summarization with user interaction.

The iNeATS system Leuski et al. (2003) was an early attempt for interactive summarization, allowing explicit control over parameters such as length, participating elements, etc. Yan et al. (2011) have studied a more *implicit* approach, attempting to discover user preferences such as topics and contexts via user clicks. Both approaches involve repeatedly updating a summary paragraph based on user feedback.

The more recent SUMMA system Christensen et al. (2014) resembles ours in supporting *hierarchical summarization*. Salient summary sentences are high in the hierarchy and further details can be discovered by drilling down into lower levels.

All of the aforementioned methods compute *extractive* summaries, which are composed of sentences from the original texts. In comparison, our *abstractive* approach has a few appealing advantages. Most importantly, this approach facilitates the construction of flexible bullet-style summaries since we are not confined to existing sentences, which may combine several atomic facts of varying saliency or require textual context. This, in turn, allows users to browse data at the level of atomic facts and avoids the need to regenerate the summary in order to incorporate user feedback.

7 Conclusion and Future Work

In this paper we presented a novel system for the interactive exploration of abstractive summary information. Our system builds on the Open Knowledge Representation Wities et al. (2017) for consolidating the information of multiple texts, and produces a summary that fully captures this information. The interactive UI allows focusing on the most salient facts as well as gradually obtaining further details via different interaction modes. Our usability studies provide supportive evidence for the usefulness of our approach.

Our results shed light on a few important directions for future research. In general, our interactive abstractive method should be ported to other domains and types of corpora. E.g., while in the case of news tweets, sentence ordering was done along a timeline, the ordering of consolidated summary sentences may in general be a non-trivial task. Further, our approach for summary sentence generation can be enhanced, e.g., by using machine learning techniques to select the best representative sentences. For evaluation, we will design tests adequate for assessing the quality of an interactive

summary, and use them in a more extensive user study that will incorporate a fully automated pipeline (i.e., an OKR parser).

Acknowledgments

This work was supported in part by grants from the MAGNET program of the Israeli Office of the Chief Scientist (OCS); the German Research Foundation through the German-Israeli Project Cooperation (DIP, grant DA 1600/1-1); by Contract HR0011-15-2-0025 with the US Defense Advanced Research Projects Agency (DARPA); by the BIU Center for Research in Applied Cryptography and Cyber Security in conjunction with the Israel National Cyber Bureau in the Prime Ministers Office; and by the Israel Science Foundation (grant No. 1157/16).

References

John Brooke. 1996. *SUS - "A quick and dirty" usability scale. Usability evaluation in industry*. CRC Press.

Janara Christensen, Stephen Soderland, Gagan Bansal, and Mausam. 2014. Hierarchical summarization: Scaling up multi-document summarization. In *ACL*.

Anton Leuski, Chin-Yew Lin, and Eduard H. Hovy. 2003. iNeATS: Interactive multi-document summarization. In *ACL*.

Wei Li, Lei He, and Hai Zhuge. 2016. Abstractive news summarization based on event semantic link network. In *COLING*.

Fei Liu, Jeffrey Flanigan, Sam Thomson, Norman M. Sadeh, and Noah A. Smith. 2015. Toward abstractive summarization using semantic representations. In *HLT-NAACL*.

Arnold M. Lund. 2001. Measuring usability with the USE questionnaire. *STC Usability SIG Newsletter*, 8(2).

Jakob Nielsen. 1993. *Usability engineering*. Academic Press.

Marco Rospocher, Marieke van Erp, Piek T. J. M. Vossen, Antske Fokkens, Itziar Aldabe, German Rigau, Aitor Soroa, Thomas Ploeger, and Tessel Bogaard. 2016. Building event-centric knowledge graphs from news. *J. Web Sem.*, 37-38.

Rachel Wities, Vered Shwartz, Gabriel Stanovsky, Meni Adler, Ori Shapira, Shyam Upadhyay, Dan Roth, Eugenio Martinez Camara, Iryna Gurevych, and Ido Dagan. 2017. A consolidated open knowledge representation for multiple texts. In *LSDSem, EACL*.

Rui Yan, Jian-Yun Nie, and Xiaoming Li. 2011. Summarize what you are interested in: An optimization framework for interactive personalized summarization. In *EMNLP*.