News Timeline Generation: Accounting for Structural Aspects and Temporal Nature of News Stream

Mikhail Tikhomirov and Boris Dobrov

Lomonosov Moscow State University, Moscow, Russia
tikhomirov.mm@gmail.com, dobrov_bv@srcc.msu.ru

Abstract. The number of news articles that are published daily is larger than any person can afford to study. Correct summarization of the information allows for an easy search for the event of interest. This research was designed to address the issue of constructing annotations of news story. Standard multi-document summarization approaches are not able to extract all information relevant to the event. This is due to the fact that such approaches do not take into account the variability of the event context in time. We have implemented a system that automatically builds timeline summary. We investigated impact of three factors: query extension, accounting for temporal nature and structure of news article in form of inverted pyramid. The annotations that we generate are composed of sentences sorted in chronological order, which together contain the main details of the news story. The paper shows that taking into account the described factors positively affects the quality of the annotations created.

Keywords: Timeline summarization · Extractive summarization · Multi-document summarization · Information retrieval

1 Introduction

Due to the explosive growth of the amount of content on the Internet, the problems of extraction and automatically summarizing useful information in the incoming data stream arises. One of such problems is the summarization of news articles on an event. The news story - is a set of news reports from various sources dedicated to describing an event. Such problems are often investigated and solved by news aggregators, for example, Google.News[1] or Yandex.News.[2] This is due to the fact that to work with such problems the researcher needs a huge and diverse collection of news articles.

The typical “lifetime” of the news story (the time of active discussion of the event) is usually a day or two, but not all events are so short. Some news stories have a “history” in the form of a set of previous events that occurred at different moments and are more or less related to each other. Existing multi-document summarization approaches do not take into account the fact that the context, actors, geography and other event properties can vary over time.

---

1 news.google.com.
2 news.yandex.ru.
The fact that journalists are returning to the same events, for example, with the appearance of new data, indicates that such events are important for the society. The need for a brief summary of the event raises the problem of forming a “timeline summary”. Timeline summary is a type of multi-document summary, containing the essential details of the subject matter under discussion. The construction of such annotations is a complex task, performed by journalists or analysts manually. This implies that the automation of such a process is a urgent problem.

In this paper we consider challenges and solutions for the automatic generation of temporal summaries. We consider this problem as a multi-document summarization on a query over a representative collection of news documents. The query in this case is the text of the news message. The situation corresponds to the scenario when a user would like to receive a timeline summary after reading the news document. The result should be a time-ordered list of descriptions of the key sub-events related to main event. The result consists of parts of existing sentences, since our solution refers to extractive summarization approaches.

A system was developed to automate the timeline summarization process. Experiments were conducted over a collection of 2 million Russian news for the first half of 2015. Three new factors were investigated to improve the results of constructing a timeline summary: query extension using pseudo-relevance feedback, accounting for the timing characteristics of news stories and the structure of the inverted pyramid.

This is a follow-up study of timeline summarization problem reported in previous paper [25]. In this study, we expanded the collection of standard annotations three-fold. The evaluation process was improved by dividing the collection into a training and test parts. An optimization module was added for fitting the configurations. As a result, substantial progress was achieved. Taking into account the structure of the inverted pyramid showed a significant increase in the values of metrics, which was not achieved in the previous article.

2 Related Work

2.1 Automatic Text Summarization Problem

Currently, there are quite a number of methods for automatic text summarization [3]. Some methods that use large linguistic ontologies [12, 15], that may be automatically supplemented during the analysis. Other methods are based on the statistical properties of texts [16] or machine learning [13].

During the generation of the annotations, the following problems occur [3, 7, 11]:

- Ensuring the completeness of the presentation of information, including the most up-to-date information.
- Decreasing of redundancy in the information provided.
- Ensuring the coherence and understandability of the information provided.

To ensure the completeness of the resulting annotation, it is often necessary to find links between sentences or documents [20].
To determine the redundancy in the generated annotations, various measures of similarity between sentences are used. One of the most common approaches is clustering - the selection of content groups of sentences [6]. Another approach to reduce redundancy is to compare candidate sentence with sentences that have already been included in the summary and to evaluate novel information. Example of such approach is the Maximal Marginal Relevance (MMR) [2].

The problem of ensuring the coherence of information in the summary arises both in the methods of generating the annotation [18, 19], and in the methods of evaluation, because in order to assess the connectivity and linguistic qualities of the annotation, it is necessary to perform a manual evaluation.

2.2 Timeline Summary

The problem of timeline summary construction has a number of differences from the standard summarization problem. For example, the temporal nature of events must be taken into account [9]. Also, to ensure completeness of the information provided, it is required to find documents from all sub-events of the topic under consideration.

When constructing timeline summary, data processing is mainly carried out over huge collections. In such collections, most of the information is not relevant to the user’s request. This problem can be solved by using clustering methods [10, 14]. But the clustering methods have some issues. First, such a task should be solved many times over huge collections of documents, which affects the response time of the system. Secondly, the degree of closeness can be significantly smaller with standard measures of similarity for documents that describe far-in-time but related events. And, of course, it is required to identify the most characteristic objects [1, 9], for example, taking into account the structural features of the flow of documents [5, 8].

3 Statement of the Problem

3.1 General Description

The problem of constructing a timeline summary is a query-oriented. In the most general case, the user has a news document as a query. So further this problem will be considered as a problem of automatic creation of a summary on a query in the form of a text document. The output of the system is an annotation of \( n \) sentences. The connectivity between the sentences in this paper is not required. Figure 1 provides an example of a possible summary about the conflict on cemetery taken from the Interfax website.\(^3\)

The aim of the work is to study the influence of various factors on the quality of the annotation.

---

\(^3\) http://www.interfax.ru/story/215.
The case of the riots in the Khovansky cemetery is sent to the court

Participants in the riots in the Khovansky cemetery were finally charged

The investigation decided not to impute the murder to the defendants of the case of the fight at the Khovansky cemetery

The court dismissed the complaint about the extension of the arrest of the defendant in the case of the fight at the Khovansky cemetery

For the defendants, the case of the fight in the cemetery will seek the status of victims

The police will continue to search for six suspects in the case of the fight at the Khovansky cemetery

Fig. 1. Timeline summary part about conflict at cemetery.

3.2 Mathematical Statement of the Problem

The problem described above can be formalized in the following way. Let $Q = \{q_1, q_2, \ldots, q_m\}$ be a set of queries and an associated set of reference annotations $D_g = \{D_{q_1}^g, D_{q_2}^g, \ldots, D_{q_m}^g\}$ be an associated set of reference annotations. The system generates a set of summary $D_A = \{D_{q_1}^A, D_{q_2}^A, \ldots, D_{q_m}^A\}$ in response to queries $Q$ by algorithm $A$. Then the problem is reduced to maximizing the following functional:

$$\sum_{i=1}^{\mid Q \mid} M(D_{q_i}^A, D_{q_i}^g) \rightarrow \max \quad (1)$$

where $M$ is the proximity function between the annotations. Optimization is carried out for all parameters of the algorithm.

4 Approach

4.1 Collection Processing

As mentioned earlier, the input collection contains 2 million news articles. It is not possible to work directly with such amount of information, therefore, it was decided to interact with the collection through a search engine. Search engine allows:

- Get a list of documents by text request.
- For a given document from collection, get the basic information: text, index, meta-information.
4.2 Studied Features

In this paper the following factors were investigated:

- Query extension strategy.
- Accounting for the temporal nature of news stories.
- Accounting for the structure of a news article in the form of an inverted pyramid.

4.3 Query Extension Strategy

Information that can be obtained from a query document is basically not enough to effectively build this type of annotation. This fact is a consequence of the fact that most news articles are not a general description of the event, but a discussion of some particular incident or fact.

To avoid this problem, it is necessary to use the query extension techniques. The developed algorithm uses the idea of pseudo-relevance feedback, which is widely used in information retrieval problems [21]. For the query-document, the algorithm includes the following steps:

1. The most significant K-terms are chosen on the basis of tf-idf weights forming thus the first-level query.
2. Further on the basis of the first-level query documents are retrieved.
3. The extracted cluster of documents is analyzed to find the most important terms forming thus the second-level query:
   a. For each document, the most significant T terms are considered.
   b. For each term, it is calculated how often it was in the top T terms on all cluster.
   c. The list of terms is sorted by frequency, the best M terms are selected.
4. Steps 2–3 are repeated (A double query extension process that forms a third-level query).
5. Output of the algorithm is a vector of N terms representing to some extent the semantics of the input document.

Note that K, T, M, N are parameters of the algorithm and they must be configured. As an example of the work of the query extension module, consider the algorithm steps on a news article about the terrorist attack in Paris (Table 1).

The table shows that a higher-level query has more significant terms for this event.

Table 1. Query extension algorithm stages example.

| Entity name            | Content                                                                 |
|------------------------|-------------------------------------------------------------------------|
| **Initial doc**        | President François Hollande of France called it a display of extraordinary “barbarism” that was “without a doubt” an act of terrorism. According to the latest information, as a result of shooting, 11 people were killed, four more are in critical condition. ... |
| **First-level query**  | Posten, Jyllands-posten, Jyllands, Herbo, Charlie, Hollande             |
| **Second-level query** | Reprint, Scandal, Weekly, Caricature, Hollande, Satirical, Terrorist act, Charlie, Herbo |
| **Third-level query (double query extension)** | Journal, Muhammad, Satirical, Attack, Prophet, Terrorist act, Paris, Caricature, Hollande, Herbo, Charlie |
4.4 Temporal Nature of News Stories

Since any event depends on time, the content of publications and their number also depend on the time. As an example, Fig. 2 shows a graph of the time dependence of publications on the “Earthquake in Nepal” event.

![Graph showing the time dependence of publications on the “Earthquake in Nepal” event.](image)

**Fig. 2.** Dependence of the number of publications per day for an event

To take into account this factor, for the set $D$ of extracted documents the following procedure is undertaken:

1. The entire timeline of the event is divided into days with labels $T = \{t_1, t_2, \ldots, t_n\}$.
2. Each document receives a label from $T$ based on the publication date $D_t$.
3. Documents published on days with a number of publications less than the $NDoc_{tr}$ threshold (2) are discarded.

$$NDoc_{tr} = 0.2 \times MEAN_{top,3}(D) \quad (2)$$

4. The output is a sorted list of collections $C = \{C_{t_1}, C_{t_2}, \ldots, C_{t_n}\}$, where each collection $C_{t_i}$ contains only documents with the label $t_i$.

4.5 Inverted Pyramid

The strategy of writing a high-quality news article often relies on the structure of the “inverted pyramid” (Fig. 3). The greatest interest is the upper and lower parts of the pyramid:

- The upper part contains the most concentrated information about the event under discussion.
- The lower part may contain references to important related events in the past.
This structure is taken into account in two ways:

1. Inter-document feature based on the graph approach.
2. Intra-document feature, which increases the weight of sentences located in the upper and lower parts of the inverted pyramid.

**Inter-Document Feature.** This feature is taken into account in the following way:

1. For a set of documents $D$, a similarity matrix between the upper and lower parts of the documents is constructed. If the specified similarity threshold is exceeded, it is considered that there is a link between the documents $D_i$ and $D_j$.
2. The importance of documents is calculated by using the LexRank algorithm over the constructed graph [4].
3. For documents whose weight is greater than a certain threshold, the previously described procedure for expanding the query is performed.

As a result, the output is a ranked list of documents $D$ and a set of $Q_D$ new queries, which further, together with accounting for the temporal nature of the news story, will help in sentence ranking algorithm. Among other things, document weights will also be taken into account in the ranking functions.

**Intra-Document Feature.** To this feature into account for the following procedure is undertaken: during the ranking of sentences, the weight of the sentence is multiplied by a coefficient that lowers the weight of sentences in the middle of the document.

Also, after described inter-document procedure, all constructed extended queries $Q_D$ are mapped to $t_i$ labels from $T$ (Fig. 4).
4.6 Similarity of Sentences

At various stages of the algorithm, there are a number of points where the measure of
closeness between sentences is calculated. For these purpose a cosine measure of
similarity (3) is used in all cases.

$$Sim_{cos}(S_i, S_j) = \frac{(S_i, S_j)}{|S_i||S_j|}$$  \hspace{1cm} (3)

The choice of representation of a sentence plays an important role for calculating
similarity. In this article we used the standard tf-idf representation. But to calculate the
similarity between sentences when searching for links between documents, word2vec
[24] representation was used. To achieve this, the resulting sentence vector is
represented as a weighted mean of word2vec word representations. Weighing was carried
out by tf-idf.

Word2vec model was trained on the entire collection of 2 million news articles. During
preprocessing removal of stop words and lemmatization were applied. The
width of the window was chosen to be 5, and the length of the vector was 100.

4.7 Sentence Ranking Module

This module deals with the ranking of sentences. The ranking is a modified version of
the MMR algorithm – MMRT (4) taking into account all the factors described in
Sect. 4.2:

$$MMRT_{st_i} = INC_{st_i} - DEC_{st_i}$$  \hspace{1cm} (4)

where \(INC_{st_i}\) – is a term describing the positive part of the formula, which depends on
the similarity of the sentence to the query, the weight of the document from which the
sentence is taken, and the sentence number in the document.

$$INC_{st_i} = (1 + \alpha * I_i) * \gamma * \lambda * Sim(Q', S_i^t)$$  \hspace{1cm} (5)

$$\gamma = 1 - 0.5 * \sin \left( \frac{i * \pi}{|D_s|} \right)$$  \hspace{1cm} (6)
The parameters \( a \) and \( k \) are configurable parameters of the algorithm, \( I_i \) – is document weight \( D_s \), which includes a sentence under the index \( i \), \( S'_i \) – is estimated sentence under the index \( i \) with label \( t \), \( Q' \) – query for this time label, \( \gamma \) – multiplier, which reduce the weight of sentences from the middle of the document.

\( DEC_{st} \) is the penalty term. It depends on the similarity to the already extracted sentences:

\[
DEC_{st} = (1 - \lambda) \times \max_{S_j \in S} \text{Sim}(S_j, S'_t)
\]  

where \( S_j \) is one of the extracted sentences, \( S \) is the set of all already extracted sentences.

Processing of sentences occurs in chronological order with a restriction on the maximum number of sentences per day.

4.8 System Diagram

The features described in Subsect. 4.2 are realized at various stages of the system. The general scheme of the algorithm is shown on Fig. 5.

![Diagram](diagram.png)

**Fig. 5.** Working scheme.
5 Evaluation

5.1 Metrics for Evaluation

The system was evaluated using several metrics: ROUGE-1, ROUGE-2, and Sentence Recall $R_{sent}$:

\[
ROUGE = N = \frac{|N_A \cap N_g|}{|N_g|}
\]

where $N_A$ is the set of n-grams for the constructed annotations, $N_g$ is the set of n-grams for the reference (gold) annotations.

\[
R_{sent} = \frac{|S_A \equiv S_g|}{|S_g|},
\]

where $S_A$ is the set of sentences from the constructed annotations, $S_g$ is the set of sentences from the reference annotations. Operator $\equiv$ denotes the following: the result of the $S_A \equiv S_g$ is a subset of $S_A$ such that their semantic equivalent is present in $S_g$.

5.2 Data Preparation

Since a test set of annotations is required for evaluating procedure, in the course of the research, timeline summaries were manually prepared. The procedure for the formation of such a collection was as follows:

1. At the first stage with the help of Wikipedia there high-profile events were selected, which were actively covered in the press for the beginning of 2015.
2. Further, for most of the events on the site “Interfax”, the search for the corresponding story was carried out. On the basis of documents corresponding to the story, a timeline summary was created.
3. If there is no corresponding story on the “Interfax”, the materials were studied on the topic and a timeline summary was created on the basis of the documents read.

As a result, 45 annotations on 15 news stories were created (Table 2).

5.3 Optimization of Algorithm Parameters

Since the system contains a large number of parameters (total 23 parameters), some of which are presented in Table 3, there was a need to optimize the choice of the values of these parameters.

To achieve this, the entire collection of the reference annotations was divided into train and test parts with the ratio 2 to 1. Further, the functional (1) was implemented in Python using an open hyperopt [22] package based on machine learning. This package uses the technique of Sequential model-based optimization (SMBO) [23] for the parameters selection. The parameters were trained on the training part. After that, the final evaluation of the configurations took place on the test part.
In order to evaluate the contribution of the considered features, a fitting and evaluation of the following 6 configurations was made:

1. **baseline** – a simple approach to summarization, without taking into account the factors considered, using MMR as ranking algorithm.

### Table 2. News stories on which the reference annotations are made.

| Story ID | Description |
|----------|-------------|
| Story 1  | The flight of the space probe DAWN to Ceres |
| Story 2  | Fire in Khakassia |
| Story 3  | Terrorist act in Paris |
| Story 4  | The plane crash in Taiwan |
| Story 5  | Earthquake in Nepal |
| Story 6  | Fire on the Orel submarine. |
| Story 7  | Attack on the synagogue in Copenhagen |
| Story 8  | Fire in the mall «Admiral» |
| Story 9  | Terrorist act in Sydney |
| Story 10 | Ice Hockey Championship |
| Story 11 | The case of Svetlana Davydova |
| Story 12 | The Murder of Boris Nemtsov |
| Story 13 | Dangerous coronavirus in Korea |
| Story 14 | Protest in Yerevan |
| Story 15 | The failure of the spaceship “Progress” |

### Table 3. Some system parameters.

| Parameter name       | Description                                                                 | Opt. value |
|----------------------|-----------------------------------------------------------------------------|------------|
| KeepL                | The number of lemmas to choose when building a first-level query             | 6          |
| DocCount             | The maximum number of documents retrieved using a search engine              | 400        |
| QuerySize            | The size of the resulting expanded query                                     | 14         |
| TopLemms             | The number of the most significant lemmas extracted in the work of the algorithm for constructing an extended query | 18         |
| MinLinkScore         | The minimum value of similarity between the top and bottom of the documents to identify the reference | 0.5        |
| Lambda               | The value of the parameter $\lambda$ for MMRT                               | 0.58       |
| Alpha                | The value of the parameter $\alpha$ for MMRT                                | 0.45       |
| MaxDailyAnswerSize   | The maximum number of sentences per day                                      | 15         |
| Doc Boundary         | Threshold of the importance of documents for building new queries           | 0.61       |

### 6 Results

In order to evaluate the contribution of the considered features, a fitting and evaluation of the following 6 configurations was made:

1. **baseline** – a simple approach to summarization, without taking into account the factors considered, using MMR as ranking algorithm.
2. **query-ex** – adding a query extension strategy feature to baseline (Sect. 4.3), but without double query extension.

3. **double-ex** – *query-ex* + double query extension (Sect. 4.3).

4. **temporal** – *double-ex* + accounting for the temporal nature of news stories (Sect. 4.4).

5. **importance** – *temporal* + accounting for the structure of a news article in the form of an inverted pyramid, when tf-idf representation is used (Sect. 4.5).

6. **w2v-imp** – *importance*, but using w2v for computing sentence similarity when accounting for the structure of a news article (Sect. 4.6).

The result of evaluation of the configurations can be seen in Table 4. This table shows that each of the features considered gives a positive contribution to the quality of generation of timeline summary. As an example of the final annotation, one can consider a fragment of the annotation on the previously mentioned incident of the crash in Taiwan in Table 5.

### Table 4. Evaluation results.

|        | $R_1$ | $R_2$ | $R_{sent}$ |
|--------|-------|-------|------------|
| baseline | 0.320 | 0.127 | 0.162      |
| query-ex | 0.326 | 0.133 | 0.175      |
| double-ex | 0.356 | 0.176 | 0.233      |
| temporal | 0.384 | 0.178 | 0.240      |
| importance | 0.399 | 0.176 | 0.246      |
| w2v-imp | **0.403** | **0.181** | **0.254** |

### Table 5. The generated timeline summary fragment about the plane crash in Taiwan.

| Date       | Sentence                                                                                                                                 |
|------------|------------------------------------------------------------------------------------------------------------------------------------------|
| 11.02.2015 | Transasia Airways will pay relatives of victims of a plane crash in Taiwan for 470 thousand                                              |
| 11.02.2015 | The tragedy in Taiwan, one-fifth of the pilots of Taiwan’s Transasia airline have not passed the proficiency test                           |
| 12.02.2015 | Rescuers completed the search operation for victims of the crash of an airline to Transasia Airways, which crashed on February 4 in Taiwan |
| 01.07.2015 | Crew crashed in Taiwan aircraft Transasia Airways shut off the engines after a loss of power                                              |
| 02.07.2015 | The Transasia plane crashed on February 4 in Taiwan, because the pilot accidentally turned off the running engine when the second engine stalled |
7 Conclusions and Future Work

In this article we presented an approach for building a timeline summary. The conducted research shows that the problem of constructing the timeline summary differs from the standard MDS problem. The effectiveness of using the following features was shown:

- Query extension strategy.
- Accounting for the temporal nature of news stories.
- Accounting for the structure of a news article in the form of an inverted pyramid.

Extending the query, as expected, has a positive effect on the event representation discussed in the document. But the interesting fact is that re-extension the query (double query extension) has a much greater effect. This is because the documents that are retrieved on the first-level query are not sufficient for a good presentation of the event.

The fact that accounting for the temporal nature of news stories improves the quality of the annotation is an obvious consequence of the fact that news stories and events have temporal characteristics.

Taking into account the structure of the inverted pyramid gives an improvement. Increase the values of metrics on the w2v-imp configuration means that the correctness of the recognized links between the documents plays a significant role. This fact raises challenges for future research.

Using structural features of news articles make it possible to obtain information, the use of which can significantly improve the quality of generated annotations.

References

1. Binh, T.G., Alrifai, M., Quoc Nguyen, D.: Predicting relevant news events for timeline summaries. In: Proceedings of the 22nd International Conference on World Wide Web, pp. 91–92. ACM (2013)
2. Carbonell, J., Goldstein, J.: The use of MMR, diversity-based reranking for reordering documents and producing summaries. In: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 335–336. ACM (1998)
3. Dang, H.T.: Overview of DUC 2006. In: Proceedings of the Document Understanding Workshop, Presented at HLT-NAACL 2006 (2006). http://duc.nist.gov/pubs/2006papers/duc2006.pdf
4. Erkan, G., Radev, D.R.: Lexrank: graph-based lexical centrality as salience in text summarization. J. Artif. Intell. Res. 22, 457–479 (2004)
5. Hu, P., Huang, M.L., Zhu, X.Y.: Exploring the interactions of storylines from informative news events. J. Comput. Sci. Technol. 29(3), 502–518 (2014)
6. Radev, D., Jing, H., Budzikowska, M.: Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. In: Proceedings of the 2000 NAACL-ANLP Workshop on Automatic Summarization, Seattle, pp. 21–30 (2000)
7. Radev, D., McKeown, K., Hovy, E.: Introduction to the special issue on summarization. Comput. Linguist. 28(4), 399–408 (2002)

8. Shahaf, D., Guestrin, C.: Connecting two (or less) dots: discovering structure in news articles. ACM Trans. Knowl. Discov. Data (TKDD) 5(4), 24–54 (2012)

9. Tran, G., Alrifai, M., Herder, E.: Timeline summarization from relevant headlines. In: Hanbury, A., Kazai, G., Rauber, A., Fuhr, N. (eds.) ECIR 2015. LNCS, vol. 9022, pp. 245–256. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16354-3_26

10. Yan, R., et al.: Evolutionary timeline summarization: a balanced optimization framework via iterative substitution. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, Beijing, China, 24–28 July 2011, pp. 745–754. ACM (2011). https://doi.org/10.1145/2009916.2010016

11. Wu, Z., Lei, L., Li, G., Huang, H., Zheng, C., Chen, E., Xu, G.: A topic modeling based approach to novel document automatic summarization. Expert Syst. Appl. 84, 12–23 (2017)

12. Hennig, L., Umbrath, W., Wetzker, R.: An ontology-based approach to text summarization. In: Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, vol. 3, pp. 291–294 (2008)

13. Nallapati, R., Zhou, B., Gulcehre, C., Xiang, B.: Abstractive text summarization using sequence-to-sequence RNNs and beyond. arXiv preprint arXiv:1602.06023 (2016)

14. Wei, T., Lu, Y., Chang, H., Zhou, Q., Bao, X.: A semantic approach for text clustering using WordNet and lexical chains. Expert Syst. Appl. 42(4), 2264–2275 (2015)

15. Allahyari, M., Pouriyeh, S., Assefi, M., Saeedi, S., Trippe, E.D., Gutierrez, J.B., Kochut, K.: Text summarization techniques: a brief survey. arXiv preprint arXiv:1707.02268 (2017)

16. Rush, A.M., Chopra, S., Weston, J.: A neural attention model for abstractive sentence summarization. arXiv preprint arXiv:1509.00685 (2015)

17. Hertzfeld, A.: Introducing Google News Timeline. https://news.googleblog.com/2009/04/introducing-google-news-timeline.html. Accessed 10 Jan 2018

18. Christensen, J., Mausam, S.S., Soderland, S., Etzioni, O.: Towards Coherent Multi-Document Summarization. In: HLT-NAACL, pp. 1163–1173 (2013)

19. Nishikawa, H., Arita, K., Tanaka, K., Hirao, T., Makino, T., Matsuo, Y.: Learning to generate coherent summary with discriminative hidden semi-Markov model. In: COLING, pp. 1648–1659 (2014)

20. Barzilay, R., Elhadad, M.: Using lexical chains for text summarization. In: Advances in Automatic Text Summarization, pp. 111–121 (1999)

21. Jiang, L., Mitamura, T., Yu, S.I., Hauptmann, A.G.: Zero-example event search using multimodal pseudo relevance feedback. In: Proceedings of International Conference on Multimedia Retrieval, p. 297 (2014)

22. Bergstra, J., Komer, B., Eliasmith, C., Yamins, D., Cox, D.D.: Hyperopt: a python library for model selection and hyperparameter optimization. Comput. Sci. Discov. 8(1), 014008 (2015)

23. Hutter, F., Hoos, Holger H., Leyton-Brown, K.: Sequential model-based optimization for general algorithm configuration. In: Coello, C.A. (ed.) LION 2011. LNCS, vol. 6683, pp. 507–523. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25566-3_40

24. Goldberg, Y., Levy, O.: word2vec Explained: deriving Mikolov et al.’s negative-sampling word-embedding method. arXiv preprint arXiv:1402.3722 (2014)

25. Tikhomirov, M.M., Dobrov, B.V.: Using news corpora for temporal summary formation (in Russian). In: Selected Papers of the XIX International Conference on Data Analytics and Management in Data Intensive Domains (DAMDID/RCDL 2017), CEUR Workshop Proceedings, Moscow, Russia, vol. 2022, pp. 165–171 (2017)