On Semi-Automatic Creation of Dataset for Multi-Document Automatic Summarization of News Articles and Forum Threads

Volodymyr Taranukha [a], Tetiana Horokhova [b] and Yaroslav Linder [a]

[a] *Taras Shevchenko National University of Kyiv, Volodymirska st., 64, Kyiv, 01033, Ukraine*
[b] *Borys Grinchenko Kyiv University, Bulvarno-Kudriavska st., 18/2, Kyiv, 04053, Ukraine*

Abstract

The problem of semi-automatic dataset creation for multi-document summarization and forum threads summarization is analyzed. Aspects specific to Slavic languages are underlined. Dedicated algorithms for this purpose were designed and tested. Due to not smooth nature of the optimization problem genetic algorithms were suggested. Some new and interesting results are received.

Keywords [1]

Automatic summarization, multi-document summarization, forum thread summarization, dataset creation

1. Introduction

The extreme proliferation of modern electronics (first and foremost - mobile phones) made electronic data sources widely available to all kinds of users. In response to this tendency multiple organizations, newspapers, forums jumped the chance to provide their own point of view, spin narrative, push advertisement, etc. This extended and enhanced data flow often takes the form of text with some images and there are too much data in the usual data flow aimed at a single person.

In this research automatic summarization is suggested as a tool to solve the problem of locating and distilling information. Among areas of application for automatic summarization two stands as more problematic:

1. Multiple document summarization.
2. Forum summarization.

On top of being more complex than a typical summarization task, there is an extra layer of problems when it comes to Slavic languages. First and foremost it's lack of a good dataset to facilitate research. The problem of multi-document summarization of news events is to provide a well-organized summary that covers an event completely while minimizing repetition. The focus and point of view of the input papers for an event may differ. Recent works in this area have tried neural models to exploit the graph structure among relations text clusters. Also, a couple of recent papers have tried neural encoder-decoder models to do multi-document summarization [1,2]. Due to the sparsity and high expense of human-written summaries, the generation of large-scale multi-document summarizing datasets for training has been hampered. There was an attempt [3] to train abstractive sequence-to-sequence models with citations and search engine results as input documents on a huge corpus of Wikipedia text. As far as efficiency goes there is a notable loss of quality in these results compared to results achieved in single-document summarization. So, a dedicated dataset to train multi-document summarization is sorely required.

The WWW discussion forums come in a variety of flavors, each with its own topic and community. User-generated content on web forums is an excellent source of information. In the case

Information technology and implementation (IT&I-2021), December 1–3, 2021, Kyiv, Ukraine
EMAIL: taranukha@ukr.net (A. 1); t.horokhova@kubg.edu.ua (A. 2); yaroslav.linder@gmail.com (A. 3)
ORCID: 0000-0002-9888-4144 (A. 1); 0000-0003-0113-8653 (A. 2); 0000-0003-1076-9211 (A. 3)

of question-and-answer sites like Quora the opening post is a question and the responses are answers to that question. The best answer in these forums may be chosen by the forum community via voting. On the other hand, there is no such thing as "the best answer" in discussion forums where people share their thoughts and experiences. Furthermore, discussion threads on a single topic might easily contain dozens or hundreds of individual posts, making it difficult to identify the important information in the thread, especially when using a mobile device to visit the forum. In this research, extractive summarization [4] is proposed to extract salient units of text from a source and then concatenate them to generate a shorter version of the discussion. Sentences are commonly utilized as summarizing units in most summarization assignments but for this assignment, it is expected that posts will be better suitable as basic units for summaries of discussion threads.

While there are many differences between both tasks (multi-document summarization and forum summarization) there are also some similarities due to the enclosed nature of articles in data sources (or posts in threads) so there is an option to exploit said similarities on top of problem-specific features. This paper describes useful elements to build the required dataset. The main point of research is to provide tools for the semi-automatic preliminary summary generation that will help to create summaries that are required for future research.

2. Related works

To assess summarizing systems, man-made reference summaries are often utilized. TIPSTER Text Summarization Evaluation Conference [5], NIST Document Understanding Conference [6], and NIST Text Analytics Conference [7] all employed benchmarks based on reference summaries.

A reference summary is a subset of text units picked from the source document for extractive summarizing, which is a researcher in this study. Depending on the task, these units might be sentences [8], utterances [9], or forum posts [10]. The first and last approaches are examined in this work. Summarization is a very subjective task: the substance of the summary varies, as does the length of the summary produced. Experts who write summaries frequently disagree on what information should be included in the summary.

To address this problem, the DUC 2005 assessment approach was established to account for diversity in human-generated reference summaries. As a result, for each of the 50 themes, at least four distinct summaries were developed. Each issue in the NIST TAC Guided Summarization Task received up to four alternative answers.

When it comes to establishing a reference, specialists writing abstractive summaries are typically asked to create a summary of a certain length for a specific document or document collection. As a result, a corpus of reference summaries usually is produced for abstractive discussion thread summarizing [11]. While abstractive summaries are not the greatest option when it comes to evaluation of extractive summaries, they can be used in conjunction with variation of ROUGE [12] metrics to make them helpful. There is ROUGE 2.0 [13] particularly designed with ability to process synonym substitution which is often used by humans in abstractive summarization tasks.

The key feature is the agreement between human experts on the content of an extractive summary. It can be measured using the percentage of common decisions and the proportions of selected and non-selected units by the experts. The agreement is then calculated in terms of effect size (number measuring the strength of the relationship between two variables in a population). Useful measure is Fleiss' κ [14]

$$\kappa = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)}, \tag{1}$$

where $\Pr(a)$ is the measured agreement (the percentage of common decisions) and $\Pr(e)$ is expected agreement based on the proportion of selected and non-selected units by the experts.

A negative κ indicates structural disagreement. If $\kappa = 0$ then there is no agreement between the experts (observed agreement is as good as random). Positive κ up to 0.2 indicates slight agreement, if

$0.2 < \kappa < 0.4$ it's fair agreement, if $0.4 \leq \kappa < 0.6$ it's moderate agreement, if $0.6 \leq \kappa < 0.8$ it's substantial agreement and if $0.8 \leq \kappa \leq 1$ it's strong agreement.

For the purpose of this research extensive search was performed but there was not a single paper found with the agreement being higher than "moderate". Among available papers, the highest scores go to single document news articles summaries ("moderate") [15]. Multi-document summarization is expectedly worse. There was no research on conversation transcripts and such but it's expected to have even worse marks there. For now, the most direct way to resolve this issue is to use voting based on the number of experts in favor of a certain segment. Also, there was found little to no good data on summarization of forum threads. All abovementioned problems are exacerbated when it comes to Slavic languages: Ukrainian and Russian languages in particular.

Recent neural network based research almost totally superseded the non-neural approach. There are papers on extractive and abstractive approach [16-19] on this matter. Though it's crucial to point out that a machine-learning based approach works well if and only if there is a sufficiently large and sufficiently comprehensive dataset. More so, some modern approaches, such as T5 [20] and GPT-3 [21] will not work without very significant investments from third parties.

So, for the purpose of this research non-neural approaches were chosen especially for extractive multi-document summarization. There were some papers on this topic both for extractive and abstractive approaches such as [22-24] albeit most of them were significantly outdated.

3. Initial analysis

Since there is no good dataset to latch to the problem was reformulated in a roundabout manner: how one can develop a feature-based method that will produce good semi-finished summaries to reduce the future workload on expert(s)?

Subsequently, this question was divided into several other questions and a number of assumptions. Basic assumptions are the following:

1. A thread is a sequence of small documents forming a discussion, with each user having a different point of view on the topic of discussion.
2. News flow is a set of documents representing different points of view corresponding to different editorial policies of news agencies. Often such a set can form a discussion if the topic stays the same during a notable period.

According to the assumptions research questions were formulated:

1. Do basic assumptions stand?
2. What is the best form to represent a preliminary multi-document news summary?
3. What is the best form to represent a preliminary thread summary?
4. What is the best length of a preliminary multi-document news summary?
5. What is the best length of a preliminary thread summary?
6. What are the characteristics of articles that are selected by humans to be included in the preliminary multi-document news summary?
7. What are the characteristics of the posts that are selected by humans to be included in the preliminary thread summary?
8. What are the major qualities important for humans in a preliminary multi-document news summary?
9. What are the major qualities important for humans in a preliminary thread summary?

To give the answer to these questions small polling was carried out among students of Taras Shevchenko National University of Kyiv. The answers were the following:

1. Yes, basic assumptions look relevant and logical.
2. For the multi-document news summary, the best is to sort articles by relative relevance to the topic and reduce each subsequent article removing irrelevant and repetitive parts.
3. For thread summary, the best way is to include whole posts if the thread is short enough. For long threads, it's useful to define the topic by first post and reduce the content of each subsequent post by removing irrelevant and repetitive parts.

4. The best length of preliminary multi-document news summary corresponds to a single screen. It was noted that it's useful to include hyperlinks to detailed articles.
5. The best length of a preliminary thread summary is from 5 to 7 posts.
6. The most important characteristic for an article to be selected into a preliminary multi-document news summary is relevance.
7. The most important characteristic for a post to be selected into a preliminary thread summary is relevance.
8. The major quality for a preliminary multi-document news summary is representativeness.
9. The major qualities for thread summary are representativeness and readability.

The answers point to the notable difference in source structure: it's much easier to obtain a readable summary from the set of articles as long as one of them serves as a basis. For thread summarization, it's much harder to get consistent (i.e. readable) summary.

These answers combined with the lack of notable dataset (especially in Ukrainian language) forced to develop a sequence of methods based on some a priori heuristics derived from a period predating the current resurgence of neural networks. From two key qualities readability looks like the one that is harder to achieve. So, readability was analyzed further.

A text differs from a set of grammatically correct sentences by a certain number of connections between the sentences that are part of it. These connections are of a different nature.

It is necessary to specify what types of connectivity are observed in the text:

- structural coherence;
- logical coherence;
- semantic closeness.

The structural coherence between the elements is most often formed by using special words or grammatical forms to connect the elements of the text. Structural coherence is defined by the following means:

- anaphora;
- elliptical structures;
- repetition of structural elements of the text, namely phrases and words;
- usage of conjunctions.

The logical coherence of the text is ensured at the level of interpretation, although it has certain syntactical features. For example, the words «якщо»("if"), «то»("then"), «інакше» ("otherwise" or "else") create logical connections in the text that are clearly different from structural connections as they were defined above. As for the task while having incomplete tools for text interpretation most of emphasis is made on structural coherence. There are several reasons for this.

First, logical coherence is based on connections such as "explanation", "cause", "consequence", and so on. The main problem lies in the fact that not all these connections and not in all cases have clear markers and the form of appropriate vocabulary or grammatical structures.

Second, logical coherence is formed by interpretation and is therefore subjective. Thus, it should be considered for each reader separately, and for this reason it is not invariant.

Third, logical coherence is formed through interpretation and therefore requires a full understanding of the text. Unfortunately, this requirement is beyond capabilities of any modern machine learning model. Thus, the algorithm is mainly based on the analysis of structural coherence, especially since it often also allows finding logical coherence, as logical coherence is often accompanied by structural coherence.

Semantic closeness is a special type of coherence and will be discussed below.

4. Algorithms to create dataset

There were two complex algorithms tested for the purpose of creation of preliminary summaries. The first was developed in IRTC IT & S in department 165 during "Pattern computer" research program [25]. Only some features and usage of the first algorithm will be described in this paper. The second algorithm was hand tailored using relatively modern instruments and is described below.

4.1. Initial processing

There is a pipeline established for the purpose of initial processing:

- subsystem of morphological analysis;
- subsystem of partial parsing;
- subsystem for simplified anaphora resolution.

Big Ukrainian dictionary (over 100,000 words) was used for morphological analysis. For the out-of-dictionary words heuristical algorithm was used [26]. The main purpose of this analysis is to find out canonical forms of words and some grammatical characteristics such as gender, case, time etc.

Having canonical forms greatly improves frequencies of text elements and also allows to process service words in correct manner. For English texts it's possible to use stemmers but for Slavic languages (Ukrainian and Russian in particular) it's notably more efficient to use dedicated morphological dictionaries. On the basis of the received morphological data the primitive syntactic analysis is carried out. Adjectives (and participles) are associated with the corresponding nouns and nouns are associated with the corresponding verbs. To do anaphora resolution morphological features are used. And only then a simple semantic closeness analysis is performed, namely: among the alternatives, the word that has the meaning that is the closest in semantic similarity to the words of the context is selected. To determine the semantic similarity, the semantic database WordNet localized to Ukrainian language [27] was used. Other parts of semantic closeness such as implied inference are ignored because they are considered too complex to analyze.

This pipeline is used in both algorithms though for the purposes of new one minor alteration were made to fit it into modern programming languages and libraries.

4.2. Important element detection

The importance of text elements is determined by how much the user is interested in them and how important they are to present the content of the text. To evaluate importance of term (word) simple tf-idf statistic is used.

The tf-idf value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general.

$$\mathrm{tf-idf_d(t)} = \mathrm{f_{t,d}} \cdot \log(\frac{\mathrm{N}}{n_t}), \tag{2}$$

where $\mathrm{f_{t,d}}$ - is the raw count of a term in a document, N - total number of documents in the selected set, $\mathrm{n_t}$ - number of documents containing term t.

This particular weighting schema promotes diversity though it's important to underline mutual influence of terms (words). It is possible to receive input document(s) with multiple synonym usages which can water down observed saliency and thus exclude important elements from summary.

To avoid this problem it's useful to calculate lexical chains [28] in texts using WordNet [29]. For the purpose of this research lexical chains were used. In order to simplify the process scores of the chains were calculated inside each text independently.

$$\mathrm{Homogenity_d(Ch)} = \frac{\mathrm{n_d\,(Ch)} - 1}{\mathrm{Length_d(Ch)}}, \tag{3}$$

where $\mathrm{n_d(Ch)}$ - number of distinct term occurrences in chain for the document, $\mathrm{Length_d(Ch)}$- length of chain (total number of occurrences of different terms in chain) in the document.

$$\mathrm{Score_d(Ch)} = \mathrm{Length_d(Ch)} \cdot \mathrm{Homogenity_d(Ch)}, \tag{4}$$

Chains were tested with quality criterion:

$$\mathrm{Score_d(Ch)} > Avg\big(\mathrm{Score_d(Ch)}\big) + 2\sigma \tag{5}$$

Where $\mathrm{Avg}\big(\mathrm{Score_d(Ch)}\big)$ is average of all scores of all chains in the particular document and σ is standard deviation.

Initially the sentences received scores based both on chain scores and on tf-idf scores.

$$\mathrm{Score_d(S)} = \sum_{\mathrm{t(S),Ch(S)}} \mathrm{tf-idf_d(t)} \cdot \mathrm{Score_d(Ch)}, \tag{6}$$

where summation includes only terms from the sentence S and if said term is also included into relevant chain.

This approach on itself penalized shorter posts or shorter articles for they often have shorter lexical chains.

4.3. Genetic algorithm

Fairly standard genetic algorithm was used in research.

The chromosome was defined as the list corresponding to sentence numbers that will be included in the final document (summary). An element having value of "True" indicates that the sentence will be included into the summary. Value of "False" corresponds to sentences that are not included into the summary. Number of values equal to "True" must not exceed summary sentence allotment.

A chromosome can mutate by randomly changing the value of a list item at random. Two chromosomes can perform cross-over in several ways:

1. Equal uniform random selection per element form parents.
2. Single point cross-over when everything before (and including) the point is taken from one parent and everything else – from another.
3. Dual point cross-over when head and tail are taken from one parent and middle part between two cut points is taken from another parent.

Each version of cross-over has own influence on performance and evaluation results.

After the cross-over is performed either padding or trimming can happen. If the summary is shorter than necessary the most salient unused sentences are included into chromosome. If the summary is longer than necessary – the least salient sentences are removed from it.

The algorithms works as following:

First generation of chromosomes is generated at random. They are places into empty List L1.
For number of generations G
    List L2 = {}
    Random mutations K times are imposed on the chromosomes.
    Mutants are placed into L2.
    For all chromosomes on the list L1
        For all chromosomes on the list L1
            A pair of chromosomes creates offsprings by cross-over
            Descendants are put into L2
    The chromosomes in L2 are sorted by rating
    N best are selected.
    If combined score of the population L1 = combined score the population L2: abort algorithm
    L1 = L2.

where G – number of generations, N - power of chromosome set, K- mutability parameter.

The rating is calculated as total sum of individual scores of terms combined with global coherence score in accordance with principles laid out in **Initial analysis**.

The first version of cross-over was implemented in the algorithm developed in Department 165. The second and third ones were implemented during this research.

5. First series of numerical experiments

For each version of cross-over experiments were performed with Ukrainian texts and the results were evaluated by hand. Each experiment consisted of 10 launches of multi-document summarization and 10 launches of forum thread summarization. During each launch of multi-document summarization 7 different articles collected from source (https://www.ukr.net/news/politics.html)

were processed. During each launch of thread summarization a thread with at least 7 posts from https://replace.org.ua/forum/9/ ("Український форум програмістів → Обговорення") was processed. Each time a score to the summary was manually assigned based on perceived performance ranging from 1 ("entirely unsatisfactory") to 5 ("good selection for future work"). The average scores and their variances are presented in Table 1. Initially it was expected that first type of cross-over will perform in the best way because it's true for many other problems that are solved using genetic algorithms. But in this experiment things went other way: first type of cross-over ended up as the worst out of three in both tasks. Also there is unexpected difference between performance of the second and third types of cross-over in both tasks. Usually it's expected to have consistent difference in performance across the board as long as the task stays the same.

Table 1
First series of evaluation results

Multi-document summarization average score	Multi-document summarization score variance	Thread summarization average score	Thread summarization score variance
3.8	0.84	3.2	1.07
4.3	0.77	3.5	0.5
4.2	0.84	3.7	0.46

6. Algorithm adjustments and second series of numerical experiments

Due to unexpected behavior of the summarization algorithm several hypotheses were put forward and tested. They mostly revolved around notions of continuity, document (post) boundaries and hidden features of human perception. For example, in most cases results of multi-document summarization retained majority of sentences from most salient (important) document and less important documents ended up in small chunks mashed together at the end of summary. Nevertheless, it does not prevent testers from giving relatively high marks to such summary. In the same time destruction (chunking) of the first post in thread summarization task was a surefire way to generate low expert score regardless to relative value (contribution) of abovementioned post to general discussion quality in thread. More so, chunking of any post carried significant negative impact on the score of generated summary while exclusion of the same post often resulted in notably milder expert score penalty. To address abovementioned issues some changes were introduced to summarization algorithm. First of all, first post in a thread was made mandatory regardless of actual contribution to summary. Second, chunking penalty was inserted into final calculations.

$$\mathrm{Penalty_d} = \frac{\mathrm{Selected_d}}{\mathrm{Total_d}}, \tag{7}$$

where $\mathrm{Selected_d}$ - number of selected sentences from the document, $\mathrm{Total_d}$ – total number of sentences in the document.

$$\mathrm{Score_d(S)} = \sum_{t,ch} \mathrm{tf-idf(t)} \cdot \mathrm{Score_d(Ch)} \cdot \mathrm{Penalty_d}\,, \tag{8}$$

It resulted in higher score if less posts were cut in pieces during general optimization. Observation of results often showed that algorithm retained initial boundaries of posts in exchange of removing some posts entirely. Also, additional changes were introduces to genetic algorithm to improve quality of summaries and speed of convergence. This approach has some similarities to chromosome reuse strategy [30] but instead of chromosome library it uses memory about ancestral behavior directly.

6.1. Chromosomes with memory

The chromosome was defined as the list corresponding to sentence numbers that will be included in the final document (summary). In comparison to standard chromosome extra field was introduced for each variable (gene). This field contains information about recent changes in variable and is used

during mutation and cross-over. There are two rules which influence behavior of chromosomes with memory field:

1. If new (mutant) chromosome is created by flipping Boolean variable that was recently in opposite state then other variable is picked for flipping.
2. If two chromosomes undergoing cross-over have too many (above certain threshold T) variables going in opposite directions (as it is indicated by respective memory fields) the cross-over does not produce offsprings. If necessary other pair of chromosomes will undergo cross-over.

For the purpose of this research extra memory field was presented by Boolean variables, but in general case memory field can be implemented as a set of integer or even real variables.

First of all this optimization is intended to boost convergence by avoiding recalculating summary quality due to algorithm clearly cycling values of the same (sub)set of variables.

While in general this optimization can be used with any kind of cross-over for the purpose of this research it was applied to dual-point cross-over as it showed the best average performance. All other versions of cross-over were removed from consideration.

6.2. Final evaluation results

The evaluation results for fourth and fifth version of preliminary summary generation are presented in Table 2.

Table 2
Second series of evaluation results

Multi-document summarization average score	Multi-document summarization score variance	Thread summarization average score	Thread summarization score variance
4.4	0.49	3.9	0.77
4.4	0.93	3.9	0.98

As it is shown in the table introduction of strict order and chunking penalty was most beneficial for thread summarization. Never the less none of the tweaks to the algorithm allowed perfect scores. Also, regardless of tweaks to the algorithm forum thread summarization is working definitely worse in comparison to multi-document summarization. On the issue of performance the results were not very conclusive. In most cases the algorithm converges notably faster, but there were some cases when the algorithm was working for the full number of generations and failed to achieve good results. This is also reflected by notable growth of score variance.

7. Conclusions and future work

For the purpose of future development of dataset(s) for evaluation of summaries some experiments were performed. It was shown that it's possible to achieve good (rated as acceptable by human experts) results with genetic algorithms for semi-automatic summary generations for multi-document summarization dataset. For the purpose of thread summarization dataset it can be beneficial to combine different approaches. Relatively high variance in scores points to some good and some poor results. It can be presumed that by picking n-best from several methods the abovementioned problems can be alleviated. The strange behavior of optimization algorithms probably emerges from the nature of the medium of thread discussions. Unlike journalists forum writers actually discuss things directly and often quote each other creating complex and convoluted chains of reasoning. More so, sometimes they don't quote directly but instead make indirect references or conclusions based on discussed subject. It makes analysis of such occasions inconvenient not only for the algorithm but also for human assistants. The future work will be centered on getting enough manpower to implement datasets for Slavic languages, first and foremost a dataset for Ukrainian language.

8. Acknowledgements

The author would like to acknowledge the following people for their contributions to the research: prof. Anisimov A.V. from faculty of Computer Sciences and Cybernetics, Taras Shevchenko National University of Kyiv for useful suggestions on the nature of natural language texts and general support; staff members of dpt. 165 of IRTC IT &S, Kyiv for libraries and support provided during this research.

9. References

[1] Y. Liu and M. Lapata, Hierarchical transformers for multi-document summarization. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, July 28 - August 2, 2019 pp. 5070–5081 URL: https://aclanthology.org/P19-1500.pdf

[2] M.Yasunaga et al., Graph-based neural multi-document summarization. Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), Vancouver, Canada, August 3 - August 4, 2017 pp. 452–462, URL: https://aclanthology.org/K17-1045.pdf

[3] Peter J. Liu et al., Generating Wikipedia by Summarizing Long Sequences. 2018. URL: https://arxiv.org/abs/1801.10198

[4] J. Xu and Durrett, G., Neural extractive text summarization with syntactic compression. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, , Hong Kong, China, November 3–7, 2019, pp. 3292–3303 URL: https://aclanthology.org/D19-1324.pdf

[5] I. Mani, et al., The TIPSTER SUMMAC text summarization evaluation. In: Proceedings of Ninth Conference of the European Chapter of the Association for Computational Linguistics, 1999 pp. 77-85. https://doi.org/10.3115/977035.977047

[6] A. Nenkova, Automatic text summarization of newswire: Lessons learned from the document understanding conference. 2005. URL: https://www.aaai.org/Papers/AAAI/2005/AAAI05-228.pdf

[7] M. El-Haj, U. Kruschwitz and C. Fox, University of Essex at the TAC 2011 MultiLingual Summarisation Pilot. 2011. URL: http://repository.essex.ac.uk/8920/1/UoEssex.proceedings.pdf

[8] H. Jing and K. McKeown, Cut and paste based text summarization. In Proceedings of 1st Meeting of the North American Chapter of the Association for Computational Linguistics. 2000. p.178-185 URL: https://aclanthology.org/A00-2024.pdf

[9] M. Li et al., Keep meeting summaries on topic: Abstractive multi-modal meeting summarization. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics July 2019. pp. 2190-2196. URL: https://aclanthology.org/P19-1210.pdf

[10] V.A. Grozin, N.F. Gusarova and N.V. Dobrenko, Feature selection for language independent text forum summarization. In: Proceedings of International Conference on Knowledge Engineering and the Semantic Web, Springer, September 2015.pp. 63-71.

[11] E. Barker et al., The SENSEI annotated corpus: Human summaries of reader comment conversations in on-line news. In: Proceedings of the 17th annual meeting of the special interest group on discourse and dialogue. September 2016. pp. 42-52. URL: https://aclanthology.org/W16-3605.pdf

[12] Kavita Ganesan, ROUGE 2.0: Updated and Improved Measures for Evaluation of Summarization Tasks Computational Linguistics, 1(1). 2006, URL: https://arxiv.org/pdf/1803.01937.pdf

[13] C.Y. Lin, Rouge: A package for automatic evaluation of summaries. In Workshop Text summarization branches out. Barcelona, Spain. July 2004. p. 74-81. URL: https://aclanthology.org/W04-1013.pdf

[14] J. L. Fleiss and J. Cohen, The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability, Educational and Psychological Measurement (1973), Vol. 33 613–619 https://doi.org/10.1177/001316447303300309

[15] M. Mitra, A. Singhal and C. Buckley, Automatic text summarization by paragraph extraction. Intelligent Scalable Text Summarization. (1997) URL: https://aclanthology.org/W97-0707.pdf
[16] K. Al-Sabahi, Z. Zuping and M. Nadher, A hierarchical structured self-attentive model for extractive document summarization (HSSAS). IEEE Access, 6, 2018. pp. 24205-24212. URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8344797
[17] K. Yao et al., Deep reinforcement learning for extractive document summarization. Neurocomputing, 284, 2018. pp.52-62. URL: https://www.researchgate.net/publication/322715462_Deep_Reinforcement_Learning_for_Extractive_Document_Summarization
[18] W. Li et al., Improving neural abstractive document summarization with explicit information selection modeling. In: Proceedings of the 2018 conference on empirical methods in natural language processing, 2018, pp. 1787-1796. URL: https://aclanthology.org/D18-1205.pdf
[19] J. Tan, X. Wan and J. Xiao, Abstractive document summarization with a graph-based attentional neural model. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics July 2017. Volume 1: Long Papers. pp. 1171-1181. URL: https://aclanthology.org/P17-1108.pdf
[20] T5 URL: https://ai.googleblog.com/2020/02/exploring-transfer-learning-with-t5.html
[21] GPT-3 URL: https://openai.com/blog/gpt-3-apps/
[22] A. Haghighi and L. Vanderwende, Exploring content models for multi-document summarization. In: Proceedings of human language technologies: The 2009 annual conference of the North American Chapter of the Association for Computational Linguistics. June 2009, pp. 362-370. URL: https://aclanthology.org/N09-1041.pdf
[23] C.Y. Lin and E. Hovy, From single to multi-document summarization. In Proceedings of the 40th annual meeting of the association for computational linguistics, July 2002, pp. 457-464. URL: https://aclanthology.org/P02-1058.pdf
[24] X. Wan and J. Yang, Multi-document summarization using cluster-based link analysis. In Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval. July 2008. pp. 299-306 URL: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.222.6018&rep=rep1&type=pdf
[25] V. Gritsenko, Zakluchnuy zvit pro vukonannya DCNTP “Obrazny konpyuter” [Final report on completion of STSTP “Pattern computer”]. IRTC IT&S, Kyyiv, 2010. 44 p. URL: http://obrazcomp.irtc.org.ua/Pressa/Zvit/Zvit_OK.pdf
[26] A.V. Anisimov, A.N.Romanik, V.Yu. Taranukha, Evrisiticheskiye algoritmy dlya opredeleniya kanonicheskih form i gramaticheskih harakteristic slov [Heuristic Algorithms for Determination of Canonical Forms and Grammatical Characteristics of Words]. Cybernetics and Systems Analysis Vol.40 (2004). – Iss. 2. pp. 3-15.
[27] Z.Wu and M. Palmer, Verb semantics and lexical selection. In: 32nd. Annual Meeting of the Association for Computational Linguistics, (1994) New Mexico State University, Las Cruces, New Mexico pp. 133 –138.
[28] A. Anisimov, et al., Ukrainian WordNet: creation and filling. In: International Conference on Flexible Query Answering Systems September 2013. pp. 649-660. Springer, Berlin, Heidelberg.
[29] R. Barzilay and M. Elhadad, Using lexical chains for text summarization. Advances in automatic text summarization, 1999. pp.111-121. URL: https://academiccommons.columbia.edu /doi/10.7916/ D8086DM3/download
[30] A. Acan, and Y. Tekol, Chromosome reuse in genetic algorithms. In Genetic and evolutionary computation conference , July 2003 Springer, Berlin, Heidelberg. pp. 695-705.