

DOI: 10.15514/ISPRAS-2025-37(3)-17



SumHiS: Extractive Summarization Exploiting Hidden Structure

^{1,2} P.A. Tikhonov, ORCID: 0009-0000-5847-2586 <Tikhonov@airi.net>

³ A.O. Ianina, ORCID: 0000-0002-6822-2801 <yanina@phystech.edu>

⁴ V.A. Malykh, ORCID: 0000-0002-4508-2527 <valentin@maly.hk>

¹ Artificial Intelligence Research Institute,
6, bldg. 2, Presnenskaya Embankment, Moscow, 123112, Russia.

² Skolkovo Institute of Science and Technology,
1, Bolshoy Boulevard 30, Skolkovo Innovation Center, Moscow, 121205, Russia.

³ Moscow Institute of Physics and Technology,
9, Institutskiy Lane, Dolgoprudny, Moscow Region, 141701, Russia.

⁴ ITMO University,
49, lit. A, Kronverksky Prospekt, St. Petersburg, 197101, Russia.

Abstract. Extractive summarization is a task of highlighting the most important parts of the text. We introduce a new approach to extractive summarization task using hidden clustering structure of the text. Experimental results on CNN/DailyMail demonstrate that our approach generates more accurate summaries than both extractive and abstractive methods, achieving state-of-the-art results in terms of ROUGE-2 metric exceeding the previous approaches by 10%. Additionally, we show that hidden structure of the text could be interpreted as aspects.

Keywords: summarization; NLP.

For citation: Tikhonov P.A., Ianina A.O., Malykh V.A. SumHiS: Extractive Summarization Exploiting Hidden Structure. Trudy ISP RAN/Proc. ISP RAS, vol. 37, issue 3, 2025, pp. 237-250. DOI: 10.15514/ISPRAS-2025-37(3)-17.

Экстрактивная суммаризация с использованием скрытой структуры

^{1,2} П.А. Тихонов, ORCID: 0009-0000-5847-2586 <Tikhonov@airi.net>

³ А.О. Янина, ORCID: 0000-0002-6822-2801 <yanina@phystech.edu>

⁴ В.А. Малых, ORCID: 0000-0002-4508-2527 <valentin@maly.hk>

¹ Институт исследований искусственного интеллекта,
Россия, 123112, г. Москва, Пресненская набережная 6, стр. 2.

² Сколковский институт науки и технологий,

Россия, 121205, г. Москва, Большой бульвар 30, стр. 1, Инновационный центр Сколково.

³ Московский физико-технический институт,

Россия, 141701, Московская область, г. Долгопрудный, Институтский переулок, 9

⁴ Университет ИТМО,

Россия, 197101, г. Санкт-Петербург, Кронверкский проспект 49, лит. А.

Аннотация. Основной целью данной работы является разработка новой модели для задачи экстрактивной суммаризации текстов, которая использует скрытую кластерную структуру текста. Авторы предлагают двухэтапный подход, включающий модель ранжирования предложений и модель обнаружения скрытой структуры текста. Для воспроизводимости результатов можно было бы сделать более детальное описание гиперпараметров и процедуры обучения моделей, также было бы информативно представить примеры полученных суммаризаций.

Ключевые слова: суммаризация; обработка естественного языка.

Для цитирования: Тихонов П.А., Янина А.О., Малых В.А. SumHiS: Экстрактивная суммаризация с использованием скрытой структуры. Труды ИСП РАН, том 37, вып. 3, 2025 г., стр. 237–250 (на английском языке). DOI: 10.15514/ISPRAS–2025–37(3)–17.

1. Introduction

Summaries are important for processing huge amounts of information. A good summary should be concise, accurate and easy-to-read. However, there can be multiple variants of a perfect summary, the same idea can be conveyed with various words. Moreover, people may find different facts of the main importance, waiting for them to be present in the summary. Most automatic text summarization algorithms do not take into account different aspects of the initial texts, providing a semantically neutral interpretation. We aim to bridge the gap between summarization approaches and aspect mining. Thus, we investigate two research directions within this work: text summarization and aspect extraction.

Text Summarization. There are two main approaches to text summarization: extractive and abstractive. Extractive methods highlight the most relevant phrases or sentences in the original text to form a summary. Alternatively, abstractive methods rephrase the text into a different form, and may not preserve the original semantic content.

Usually summarization has an underlying suggestion, that one summary should fulfill every informational demand. That is not true in many cases, e.g., imagine text about fruits in general, while a person is interested exactly in apples. In that toy example the proper summary for the aforementioned person should contain maximum information about apples with some occasional references to other fruits. Such a result can be achieved with aspect extraction techniques. The aspect extraction underlying suggestion is that each document consists of several aspects.

Hidden Document Structure. Revealing hidden document structure is important for getting a concise and accurate summary. One way to do so is via aspect extraction. Each aspect may be specified by explicit words or sometimes inferred implicitly from the text. For example, in the sentence “the image is very clear” the word “image” is an aspect term. The associated problem of aspect categorization is to group the same aspect expressions into a category. For example, the aspect terms “image”, “photo”, and “picture” can be grouped into one aspect category named Image.

Hidden document structure is conventionally associated with dividing a document into multiple facets, each of which may have its own sentiment. However, the structures may relate to different textual features, e.g., topics covered in the text. In this paper we concentrate on how the discovered structures helps to make the summaries more accurate. Although we do not interpret these discovered structures as aspects.

Our Approach. We propose an extractive summarization model, that we call SumHiS (*Summarization with Hidden Structure*), which utilizes representations from BERT model [1] and uses topical hidden document structure. In this work we introduce two blocks for creating extractive summaries. First, we use contextualized representations retrieved from a pre-trained language models to rank the sentences from a document according to their importance. Second, we further filter the already ranked sentences in order to focus the summary on the facts corresponding to main discovered topics within document.

Evaluated on CNN/DailyMail dataset [2], our approach outperforms previous extractive summarization state-of-the-art in terms of ROUGE-2 [3] metric by 10%. These results demonstrate the importance of topical structure inclusion for summarization task. Furthermore, we capitalize on the power of pre-trained language models combined with document structure discovery, that makes the resulting summary to focus on the most important topics and ideas mentioned in the initial text.

To summarize our key contributions are:

- A novel extractive summarization pipeline, which combines representations from pre-trained language models and hidden document structure discovery techniques.
- Our method outperforms prior work on the CNN-DailyMail dataset by a large margin in terms of ROUGE-2 and ROUGE-L metrics and can successfully be applied to real-world applications.
- Moreover, our model outperforms abstractive models too.

2. Related work

The earliest attempts of automatic summarization focused on extractive techniques, which find words or sentences in a document that capture its most salient content. Recent works use a variety of approaches. For example, [4] proposed a novel summary-level framework MatchSum and conceptualized extractive summarization as a semantic text matching problem. The authors proposed a Siamese-BERT architecture to compute the similarity between the source document and the candidate summary. In [5] the authors rely on extractive summarizers that identify salient sentences based on positional information.

Under supervised learning conditions, aspect-level sentiment classification is typically considered a classification problem. Early works [6-8] mainly used manually designed features such as sentiment lexicon, n-grams, and dependency information. However, these methods highly depend on the quality of the designed features, which is labor-intensive. With the advances of deep learning methods, various neural models [9-11] have been proposed for automatically learning target-dependent sentence representations for classification. The main idea behind these works is to develop neural architectures that are capable of learning continuous features without feature engineering and at the same time capturing the intricate relatedness between a target and context words.

Of course, there are many works in recent years in abstractive summarization. In the work [2] authors proposed to use encoder-decoder on a huge corpus to achieve good results in the abstractive summarization task. Later in work [12] use a different type of recurrence network and obtained the state-of-the-art results. Nallapati and co-authors used word copying mechanism from the input sequence to the output, thereby solving the problem with rare words. In the paper [13] Cohan and co-authors proposed a summarization model for very long documents, like scientific articles. They use the hierarchical encoder mechanism that models the discourse structure of a document. Putra et

al. [14] proposed to use so-called topical sentence, i.e., the most important one from the article, to generate news headline.

The last-mentioned works allowed us to suggest a hidden structure usage in summarization. We chose a model which is designed to capture a hidden structure, namely extract aspects from texts. Neural attention-based aspect extraction model (ABAE) is proposed in [15]. The main idea of this work is to create a matrix of vector representations which could be used to reconstruct a sentence vector representation. It is done under assumption that there is only one main aspect which a sentence has.

In the models like MatchSum, the authors use vector BERT representations of the sentences. We decided to follow this approach, but instead of classic binary prediction whether a sentence should be included or not we chose ranking approach, allowing us to filter the sentences basing on their score. We chose recent state of the art approach in text ranking SparTerm [16]. This model is using vector representations of input texts to predict their ranking. The vector representations are produced from fine-tuned BERT model. We adopted this approach with exception of irrelevant to us term prediction task.

3. Model description

This section presents the general overview of our extractive summarization system SumHiS, its architecture and the corresponding training strategy. Our system consists of two blocks: sentence ranking model and hidden structure discovery model. The model's interaction is shown in Fig. 1. The training process of our system also consists of two phases. First, we train sentence ranking model and then we use its output representations to train a hidden structure discovery model.

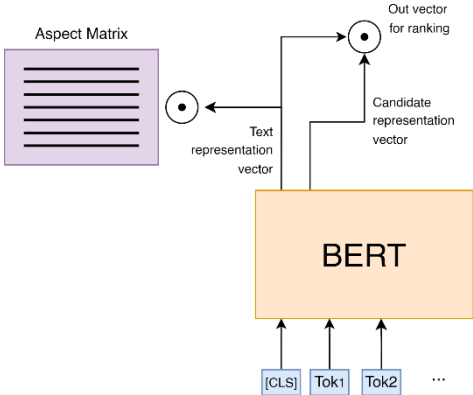


Fig. 1. SumHiS: ranking model (right) + hidden structure discovery model (left).

3.1 Ranking

We follow Term-based Sparse representations (SparTerm) setup [16] to train a ranking model. SparTerm learns sparse text representations by predicting the importance for each term in the vocabulary. Since we are working with extractive summarization, our model estimates importance of sentences instead of terms.

SparTerm represents text using BERT [1] model as follows: a text is fed into the model, and each term is embedded to a vector space. The term embeddings are averaged and used as a single text embedding. This text embedding is compared to other text embeddings thereby producing similarity scores. Similarly to SparTerm setup, we use BERT with specifically designed input. Each input is represented as a triplet (*text*, *pos_sentence*, *neg_sentence*), where *text* is a whole text of a document, *pos_sentence* is a sentence included into the golden summary, and *neg_sentence* is a sentence not included into the golden summary. The visualization of the model input is presented in Fig. 2.

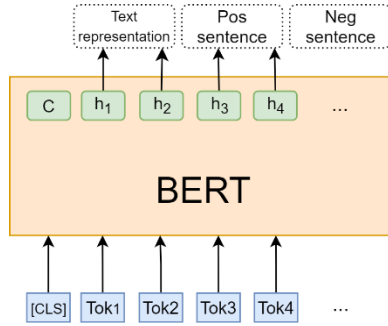


Fig. 2. Ranking model.

We aim to make a positive sentence representation as close as possible to a representation of a text and simultaneously make a representation of negative sentence as far as possible from it. Let $R = \{(t_1, s_{1,+}, s_{1,-}), \dots, (t_N, s_{N,+}, s_{N,-})\}$ denote a set of N training instances; each containing text t_i , positive candidate sentence $s_{i,+}$ and negative one $s_{i,-}$, indicating that $s_{i,+}$ is more relevant to the text than $s_{i,-}$. The ranking model is trained by optimizing a ranking objective which in our case is negative log likelihood of the positive sentence:

$$L_{\text{summ}}(t_i, s_{i,+}, s_{i,-}) = -\log \frac{e^{\text{sim}(t'_i, s'_{i,+})}}{e^{\text{sim}(t'_i, s'_{i,+})} + e^{\text{sim}(t'_i, s'_{i,-})}}$$

where $t'_i, s'_{i,+}, s'_{i,-}$ are dense representations of $t_i, s_{i,+}, s_{i,-}$ respectively, and sim denotes any similarity function. We use dot-product in our experiments.

During the training each document t_i is split into sentences, from which the triplets are generated. The output of the ranking model is an ordering of the sentences which are similar to the text summary from the closest one to the most distant one. Given this ordering one could create a summary for the text taking several top sentences.

3.2 Hidden Document Structure Discovery

Through experimentation, we found out that quality of summaries can be increased by adding information about hidden document structure.

We follow ABAE model [15] setup in order to capture hidden document structure. A sentence vector representation is considered to consist of weighted sum of several cluster representations. In case of ABAE these clusters are interpreted as aspects, while in our model we do not follow this interpretation and consider them as ordinal clusters.

The structure discovery model learns a matrix C of cluster embeddings of size $K \times n$, where K is the number of clusters, n is an embedding space size. We use an attention-like mechanism to take into account all the cluster representations and reconstruct the initial vector. We calculate each input score by calculating its dot product with each cluster embedding:

$$p_j = c_j \cdot q$$

where q is an input text vector representation, while c_j is j -th cluster embedding in the embedding matrix. Obtained scores are then normalized with softmax function, leaving us with one highest weight corresponding to the leading cluster representation for q . Next, each cluster vector is multiplied by the corresponding weight and summed up to get the output reconstructed vector o :

$$o = \sum_{j=1}^K p_j c_j$$

This output reconstructed vector is expected to be similar to the input text vector, so in order to train structure discovery model we minimize the loss function based on cosine distance:

$$L_{asp} = 1 - \frac{q \cdot o}{|q||o|}$$

Such training allows us to build a model which could represent any input vector as a sum of one leading cluster representation and several others. This model is used for filtering of a vector set, it is an ordered set of sentence representations in our case. We filter the set in the following way. Let us say that p_q is a weight for the leading cluster for the input text q . We could filter out any sentence i from the set, where:

$$p_a^i \leq \text{threshold}$$

and *threshold* could be selected arbitrarily.

4. Datasets

CNN/Daily Mail [2] is a dataset commonly used for text summarization evaluation. Human generated abstractive summary bullets were generated from news stories in CNN and Daily Mail websites as questions (with one of the entities hidden), and stories as the corresponding passages from which the system is expected to answer the fill-in-the-blank question. The authors released the scripts that crawl, extract and generate pairs of passages and questions from these websites.

All in all, the corpus has 286,817 training pairs, 13,368 validation pairs and 11,487 test pairs, as defined by their scripts. The source documents in the training set have 766 words spanning 29.74 sentences on an average while the summaries consist of 53 words and 3.72 sentences.

XSum [17] is a dataset for evaluation of abstractive single-document summarization systems. The goal is to create a short, one-sentence new summary answering the question “What is the article about?”. The dataset consists of 226,711 news articles accompanied with a one-sentence summary. The articles are collected from BBC articles (2010 to 2017) and cover a wide variety of domains (e.g., Politics, Sports, Weather, and Technology). The official split contains 204,045 (90%), 11,332 (5%) and 11,334 (5%) documents in training, validation and test sets, respectively.

4.1 Converting to Extractive Dataset

Although the datasets are originally designed for abstractive summarization, we modified them for extractive summarization using a special utility. To obtain the extractive summaries from abstractive ones we use classic concept of extractive oracle. We define the extractive oracle summaries as follows, using ROUGE metrics described below:

$$O = \operatorname{argmax}_{S \subseteq D} \operatorname{ROUGE}_N(G, S), \text{ s.t. } \ell(S) \leq 2\ell(G)$$

Here D is the set of all the sentences contained in the input document, and G is the gold (abstractive) summary for the input document. $\ell(\cdot)$ indicates the number of words in a text.

5. Experiments

5.1 Metrics

The models are evaluated with F1 variant (harmonic mean of Precision and Recall) of ROUGE-1, ROUGE-2, ROUGEL [18]. ROUGE-N is computed as follows:

$$\operatorname{ROUGE}_N = \frac{\sum_{S \in \operatorname{Ref}} \sum_{g_n \in S} \operatorname{Count}_{\operatorname{match}}(g_n)}{\sum_{S \in \operatorname{Ref}} \sum_{g_n \in S} \operatorname{Count}(g_n)}$$

Where n° stands for the length of the n-gram g_n , and $\operatorname{Count}_{\operatorname{match}}(g_n)$ is the maximum number of n-grams cooccurring in a candidate summary and a set of reference summaries Ref° .

- ROUGE-1 value measures the overlap of unigram (each word) between the computed summary and the gold summary.

- ROUGE-2 value measures the overlap of bigrams respectively.
- ROUGE-L measures the longest common subsequence between the model output and gold summary.
- Recall in the context of ROUGE means how much of the gold summary is the computed summary capturing.
- Precision answers how much of the computed summary was in fact relevant.

5.1 Baselines

We compare our model to the following models.

5.1.1 Extractive Models

MatchSum [4]: this approach formulates the extractive summarization task as a semantic text matching problem. A good summary should be more semantically similar to the source document than the unqualified summaries.

DiscoBERT [19]: the model extracts sub-sentential discourse units (instead of sentences) as candidates for extractive selection on a finer granularity. To capture the long-range dependencies among discourse units, structural discourse graphs

BertSumExt [20]: the model uses pretrained BERT with inserted [CLS] tokens at the start of each sentence to collect features for the sentence preceding it.

5.1.2 Abstractive Models

Selfmem [21]: a two-stage framework consisting of a retrieval-augmented generator and memory selector, where the generator's outputs serve as memory candidates for subsequent generation rounds in an iterative process, addressing limitations of fixed-corpus retrieval.

BRIO [22]: a model that modifies the target distribution of abstractive models from a deterministic (one-point) distribution to a non-deterministic distribution where candidate summaries are assigned probability mass according to their quality.

SimCLS [23]: a two-stage model for abstractive summarization, where a Seq2Seq model is first trained to generate candidate summaries with MLE loss, and then a parameterized evaluation model is trained to rank the generated candidates with contrastive learning.

GSum [24]: the model has two encoders which encode the source document and guidance signal, which are attended to by the decoder.

ProphetNet [25]: Transformer-based model which is optimized by n-step ahead prediction that predicts the next n tokens simultaneously based on previous context tokens at each time step.

5.2 Experimental Setup

For the summarization model, a pre-trained BERT was used (*bert-base-uncased* variation from the Transformers library [26]). Input sequence goes as follows:

`[CLS]text[SEP]sentence_1[SEP]sentence_2`

text is limited or padded to 430 tokens, while *sentence_1* and *sentence_2* are both limited to 39 tokens. *sentence_1* and *sentence_2* are filled with *pos_sentence* or *neg_sentence* randomly to force the model to not rely upon their relative ordering and use an embedded semantics.

During the evaluation, each document is split to sentences the exact same way as during the training. Each sentence is considered to be a candidate for inclusion in summary. It is fed into the model as *pos_sentence*. As *neg_sentence* we use the last sentence in a text, since we assume it is never included into the summary.

The structure discovery model is trained for two epochs, the threshold for filtering was set to 0.25.

6. Results

We compared our model with current state of the art. We denote our model *SumHiS with* and *without filtering* for the variants of the model where the hidden structure discovery model is present or not respectively (see Fig. 3).

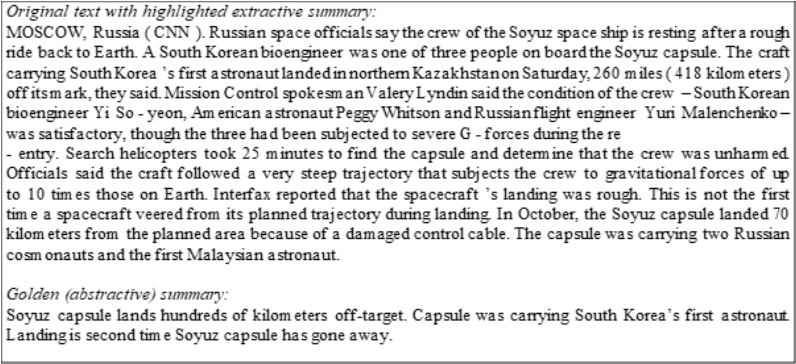


Fig. 3. Sample of SumHiS generated summary.

We evaluate the models on the CNN/DailyMail dataset in non-anonymized version. The evaluation results are presented in Table 1. One could see that our model shows the superior performance among the extractive models by the means of ROUGE-2 and ROUGE-L improving the previous results by almost 12% and 2% respectively. ROUGE-1 evaluation result for our model is 1 percent lower than state of the art result. Thus, could conclude that our model is more successful in extraction of longer sequences of tokens, while keeping the unigrams distribution close to the desired one.

Table 1. ROUGE metrics for the extractive models on CNN/DailyMail test set (non-anonymized). Best result is given in bold, second best – in italic.

Model	ROUGE-1	ROUGE-2	ROUGE-L
MatchSum [4]	44.41	20.86	40.55
DiscoBERT [19]	43.77	20.85	<i>40.67</i>
BertSumExt [20]	43.85	20.34	39.90
SumHiS (w/o filtering)	38.43	<i>28.51</i>	37.58
SumHiS (with filtering)	43.48	32.52	42.44

In addition, we compare our model with abstractive models. The results are presented in Table 2. Despite that our model is not using the generation, i.e., paraphrase ability of the language models, it shows the best results by ROUGE-2 metric outperforming the previous approaches by 10%. ROUGE-L is evaluated only 1 percent lower than state of the art result. This result is an intriguing one, since the extracted bigrams are still better fit the desired distribution than the generated ones. It is important to mention, that structure discovery has significant influence on the model output, leading to improvement by 5% in ROUGE-1 and ROUGE-L and by 4% in ROUGE-2. We also provide a sample of SumHiS output in comparison to golden summary in Table 3.

7. Analysis

The threshold in the experiments was not chosen randomly. We conducted a series of experiments resulting receiver output characteristic for the filtering classifier showed at Fig. 4). The vertical axis is true positive rate, while horizontal one is false positive rate. The value of 0.25 shows the best balance between them.

Table 2. ROUGE metrics for the abstractive & our models on CNN/DailyMail test set (non-anonymized). Best result is given in bold, second best – in italic.

Model	ROUGE-1	ROUGE-2	ROUGE-L
BRIO [22]	47.78	23.55	44.57
SimCLS [23]	46.67	22.15	43.54
GSum [24]	45.94	22.32	42.48
Ours (w/o filtering)	38.43	28.51	37.58
Ours (with filtering)	43.48	32.52	42.44

Table 3. ROUGE metrics for the XSum test set. Best result is given in bold, second best – in italic.

Model	ROUGE-1	ROUGE-2	ROUGE-L
Selfmem [21]	50.3	26.7	41.6
BRIO [22]	49.07	25.59	40.40
SimCLS [23]	47.61	24.57	39.44
GSum [24]	45.40	21.89	36.67
MatchSum [4]	24.86	4.66	18.41
BertSumExt [20]	22.86	4.48	17.16
Ours (w/o filtering)	41.94	28.80	39.97
Ours (with filtering)	43.20	30.32	41.46

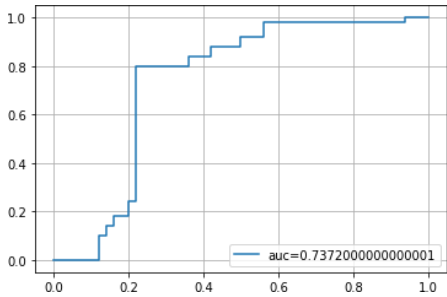


Fig. 4. True Positive Rate vs. False Positive Rate for SumHiS with different threshold values.

7.1 Ablation study

The resulting SumHiS system has several choices which we did basing on the experiment results. SumHiS system contains two models, namely ranking and structure discovery ones. The choices for these models could be questioned thus we provide the results of an ablation study in Table 4. All the results are achieved on CNN/DailyMail dataset. The metrics in this table are the variants of ROUGE, e.g., R-1-p is an abbreviation for ROUGE-1-Precision, R-2-r stands for ROUGE-2-Recall, while R-L-f means ROUGE-L F-measure variant of the metric. All the other metrics are named analogously. We provide more complete results for SumHiS with and without filtering, naming them respectively in the table. We have also tried to interpret the summarization task as binary classification problem, since it is a common approach in the field. In this setup we generate the following triplets: $(t_i, s_{i+}, 1)$ and $(t_i, s_{i-}, 0)$. The last value in a triplet is a label to predict. As a loss function, we use classic binary cross-entropy. The results of this attempt are named “SumHiS + binary loss”. One could see that such replacement of a loss function is leading to catastrophic degradation of SumHiS quality by the means of Precision and F-measure as a consequence.

Table 4. Results for different diode insertion strategies.

Model \ Metric	R-1-p	R-1-r	R-1-f	R-2-p	R-2-r	R-2-f	R-L-p	R-L-r	R-L-f
SumHiS w/o filtering	26.99	79.01	38.43	19.21	65.59	28.51	26.40	77.32	37.58
SumHiS with filtering	31.94	79.45	43.49	22.77	67.02	32.52	31.16	77.62	42.44
SumHiS + aspects	19.24	100.00	31.12	13.67	98.64	23.07	19.31	100.00	30.87
SumHiS + binary loss	17.64	100.00	28.61	12.47	91.88	21.08	17.64	100.00	28.61
Orig. BERT	18.93	100.00	30.69	13.41	98.80	22.67	18.64	100.00	30.33
BERT with filtering	19.12	99.99	31.01	13.59	98.77	23.02	19.08	99.99	30.51
BERT + aspects	18.44	92.12	29.33	12.75	78.03	21.11	18.32	91.53	29.15

We considered the original BERT model without any fine-tuning on our data for the extractive summarization. We used the following setup as for SumHiS, we average per token representations to obtain the input text representation. To make an ordering required to produce a summary compare document text vector representation t' with sentence representation s' . We use dot product as comparison function. The results for this model are denoted as Orig. BERT in the table. Interestingly, Orig. BERT model shows better performance by ROUGE-2 (R-2-f in terms of Table 4), than BERT-based BertSumExt model, although the other metrics are significantly lower for it.

Next, we applied our structure discovery model to the output of the Orig. BERT. The results are denoted as BERT with filtering. The structure discovery has improved all the metrics of Orig. BERT (not including R-1-r and R-L-r, since they were 100%). The achieved result in ROUGE-2 (R-2-f) is a new state of the art, if we are leaving SumHiS aside. Although the improvement is small, it is consistent for all metrics. This result is correlated with filtering usage with SumHiS.

At last, but not least we have experimented with our structure discovery model. It is partially following ABAE setup with two important differences: we do not used initialization for the clusters (aspects) and we do not regularize the cluster matrix. The initialization ABAE use is following: it takes vector representations of all the unique words in the training dataset; apply K -means clustering algorithm to the vectors, where K is set to be the desired number of aspects; and finally averaging all the vectors in a cluster to get its centroid vector. The centroids are used as initial values for the aspect embeddings. The regularization which is used in ABAE is orthonormal one. It is formulated as follows:

$$L_{ortho} = ||C \times C^T - I||$$

where C is an aspect matrix of size $K \times n$ and I is diagonal unit matrix of size $K \times K$. We have applied both of these techniques to our model and Orig. BERT. The results are denoted as “+ aspects” in the table. Surprisingly to us, addition of aspect filtering is lowering all the metrics for both BERT and SumHiS models. Although the aspects extracted with this method seem to be adequate, the quality of the main task of summarization is too low to consider this approach as a general one. A sample of aspects extracted by SumHiS is presented in Table 5.

Table 5. Aspects, extracted by SumHiS.

#	Aspect words
1	girl, women, teenagers, tennis, ladies, princess, kids
2	perhaps, apparently, probably, mysteriously, presumably, suddenly, supposedly
3	care, reasons, aids, purpose, irregularities, conform, attention
4	desperately, supplying, blood, terribly, abducted
5	building, built, forcing, using, saying, trying

7.2 Vector Space Analysis

We aim the model to output different vectors for positive and negative input sentences. To prove it, we calculated distances between $text'$ and $pos_sentence'$ and $text'$ and $neg_sentence'$ for every triplet in the test set. As shown in Fig. 5 (left), the distances between initial text representation and negative sentence representations are generally greater than the ones between the initial text and the positive sentence. To take into account the peaking values we performed the kernel trick (see Fig. 5, right): $(x - 0.45)^2$, where x is an initial distance between text and sentence.

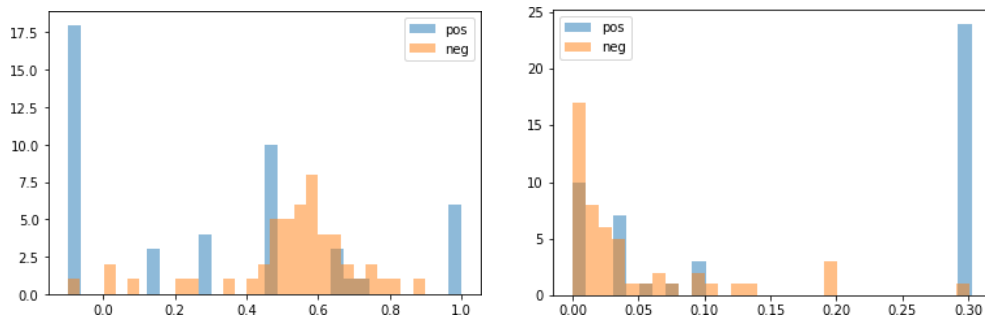


Fig. 5. Histogram of distances between initial text and positive (blue) / negative (orange) sentences. Left – without the kernel trick, right – after performing the kernel trick.

8. Conclusion

We proposed a new model for extractive summarization that uses information about hidden document structure. Our model shows state-of-the-art performance on CNN/DailyMail dataset by ROUGE-2 and ROUGE-L compared to current extractive summarization models. Moreover, it shows the best performance by the means of ROUGE-2 in comparison with abstractive models outperforming them by 10%. We showed that hidden structure in a text could be successfully used leading to significant improvements in summary generation. As for the future work, we plan to make SumHiS end-to-end trainable aggregating ranking model and structure discovery models into an integral pipeline. We are also considering to integrate structure discovery within abstractive summarization, and experiment with different structure discovery mechanisms.

References

- [1]. Devlin J., Chang M.-W., Lee K., Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.
- [2]. Nallapati R., Zhou B., Gulcehre C., Xiang B. Abstractive text summarization using sequence-to-sequence RNNs and beyond. arXiv preprint arXiv:1602.06023, 2016.
- [3]. Lin C.-Y. ROUGE: A package for automatic evaluation of summaries. In Text Summarization Branches Out, 2004, pp. 74-81.
- [4]. Zhong M., Liu P., Chen Y., Wang D., Qiu X., Huang X.-J. Extractive Summarization as Text Matching. In Proc. 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 6197-6208.
- [5]. Dong Y., Romascanu A., Cheung J. C. K. HipoRank: Incorporating hierarchical and positional information into graph-based unsupervised long document extractive summarization. arXiv preprint arXiv:2005.00513, 2020.
- [6]. Boiy E., Moens M.-F. A machine learning approach to sentiment analysis in multilingual Web texts. Information retrieval, vol. 12, no. 5, 2009, pp. 526-558.
- [7]. Kiritchenko S., Zhu X., Cherry C., Mohammad S. NRC-Canada-2014: Detecting aspects and sentiment in customer reviews. In Proc. 8th International workshop on Semantic Evaluation (SemEval 2014), 2014, pp. 437-442.

- [8]. Wagner J., Arora P., Cortes S., Barman U., Bogdanova D., Foster J., Tounsi L. DCU: Aspect-based polarity classification for SemEval task 4. In Proc. 8th International workshop on Semantic Evaluation (SemEval 2014), 2014.
- [9]. He R., Lee W. S., Ng H. T., Dahlmeier D. Exploiting document knowledge for aspect-level sentiment classification. arXiv preprint arXiv:1806.04346, 2018.
- [10]. Chen P., Sun Z., Bing L., Yang W. Recurrent attention network on memory for aspect sentiment analysis. In Proc. 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 452-461.
- [11]. Liu J., Zhang Y. Attention modeling for targeted sentiment. In Proc. 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, 2017, pp. 572-577.
- [12]. Nallapati R., Zhai F., Zhou B. SummaRuNNer: A recurrent neural network based sequence model for extractive summarization of documents. In Proc. Thirty-First AAAI Conference on Artificial Intelligence, 2017.
- [13]. Cohan A., Dernoncourt F., Kim D. S., Bui T., Kim S., Chang W., Goharian N. A Discourse-Aware Attention Model for Abstractive Summarization of Long Documents. In Proc. 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), 2018, pp. 615-621.
- [14]. Putra J. W. G., Kobayashi H., Shimizu N. Experiment on Using Topic Sentence for Neural News Headline Generation, 2018.
- [15]. He R., Lee W. S., Ng H. T., Dahlmeier D. An unsupervised neural attention model for aspect extraction. In Proc. 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2017, pp. 388-397.
- [16]. Bai Y., Li X., Wang G., Zhang C., Shang L., Xu J., Wang Z., Wang F., Liu Q. SparTerm: Learning Term-based Sparse Representation for Fast Text Retrieval. arXiv preprint arXiv:2010.00768, 2020.
- [17]. Narayan S., Cohen S. B., Lapata M. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. arXiv preprint arXiv:1808.08745, 2018.
- [18]. Lin C.-Y. ROUGE: A package for automatic evaluation of summaries. In Proc. Text Summarization Branches Out, 2004, pp. 74-81.
- [19]. Xu J., Gan Z., Cheng Y., Liu J. Discourse-Aware Neural Extractive Text Summarization. In Proc. 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 5021-5031.
- [20]. Liu Y., Lapata M. Text Summarization with Pretrained Encoders. In Proc. 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 3730-3740.
- [21]. Cheng X., Luo D., Chen X., Liu L., Zhao D., Yan R. Lift yourself up: Retrieval-augmented text generation with self-memory. *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [22]. Liu Y., Liu P., Radev D., Neubig G. BRIO: Bringing order to abstractive summarization. arXiv preprint arXiv:2203.16804, 2022.
- [23]. Liu Y., Liu P. SimCLS: A Simple Framework for Contrastive Learning of Abstractive Summarization. arXiv preprint arXiv:2106.01890, 2021.
- [24]. Dou Z.-Y., Liu P., Hayashi H., Jiang Z., Neubig G. GSum: A General Framework for Guided Neural Abstractive Summarization. In Proc. 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2021, pp. 4830-4842.
- [25]. Qi W., Yan Y., Gong Y., Liu D., Duan N., Chen J., Zhang R., Zhou M. ProphetNet: Predicting Future N-gram for Sequence-to-Sequence Pre-training. In Proc. 2020 Conference on Empirical Methods in Natural Language Processing: Findings, 2020, pp. 2401-2410.
- [26]. Wolf T., Debut L., Sanh V., Chaumond J., Delangue C., Moi A., Cistac P., Rault T., Louf R., Funtowicz M., Davison J., Shleifer S., von Platen P., Ma C., Jernite Y., Plu J., Xu C., Le Scao T., Gugger S., Drame M., Lhoest Q., Rush A. M. Transformers: State-of-the-Art Natural Language Processing. In Proc. 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Online, 2020, pp. 38-45.

Информация об авторах / Information about authors

Павел Александрович ТИХОНОВ – магистр компьютерных наук, аспирант Сколтеха, научный сотрудник AIRI. Сфера научных интересов: обработка естественного языка, генеративные модели, интерпретируемость.

Pavel Alexandrovich TIKHONOV – Master Sci. (Comp. Sci.), postgraduate student at Skoltech, researcher at AIRI. Research interests: Natural Language Processing, Generation Models, Interpretability.

Анастасия ЯНИНА – руководитель направления LLM в Wildberries. Сфера научных интересов: языковые модели, эмбединги для обработки текста, оптимизация поисковых систем маркетплейсов.

Anastasia YANINA – Head of LLM Department at Wildberries. Research interests: Language Models, Text Embeddings, Marketplace Search Optimization.

Валентин Андреевич МАЛЫХ – кандидат технических наук, руководитель исследований в области обработки естественного языка в MTS AI. Сфера научных интересов: генерация естественного языка, глубокое обучение.

Valentin Andreevich MALYKH – Cand. Sci. (Tech.), NLP Research Head at MTS AI. Research interests: Natural Language Generation, Deep Learning.

