Extractive Topical Summarization With Aspects

Anonymous ACL submission

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 topical structure and information about aspects of the text. Experimental results on CNN/DailyMail demonstrate that our approach generates more accurate summarizations than baseline methods, achieving state-of-the-art results in terms of ROUGE metric. Additionally, we show that aspect information is extremely important in extractive summarization scenario.

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: extrative and abstractive. Extractive methods highlights 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.

The summarization has an underlying suggestion, that one summary should fit to all. That is not true in many cases, e.g. a text tells a story about the fruits, while a person is interested only in apples. In that toy case the proper summary for this person should contain maximum information about the apples with some occasional references to other fruits. We suppose that we could address this issue by introducing aspect extraction techniques. The aspect extraction underlying suggestion is that each document consists of several aspects.

Aspect extraction. Aspect extraction is the task of identifying and extracting terms relevant for opinion mining and sentiment analysis, for example terms for product attributes or features. Aspects may be specified by explicit words or sometimes they can be 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. One reason why deep learning models can be helpful for this task is that, deep learning is essentially good at learning (potentially complicated) feature representations. When an aspect is properly characterized in some feature space, for example, in the hidden layers, the semantics or correlation between an aspect and its context can be learned. In other words, deep learning provides a possible approach to automatic feature engineering without human involvement.

Aspect extraction is conventionally associated with dividing a document into multiple facets, each of which may have its own sentiment.

We propose an extractive summarization model which utilizes representations from pretrained BERT enriching them with aspects retrieved from initial text with our aspect extraction model. Evaluated on CNN/DailyMail dataset (Nallapati et al., 2016), the overall approach outperforms the previous extractive summarization state-of-the-art (see Table 1) in terms of ROUGE (Lin, 2004) metric. Thus, we demonstrate the importance of aspect information while generating a summary.

2 Related Work

The earliest attempts at 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 (Zhong et al., 2020) proposed a novel summary-level framework MatchSum and conceptualized extractive summarization as a semantic text matching problem. They proposed a Siamese-BERT architecture to compute the similarity between the source document and the candidate summary. In (Dong et al., 2020) 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 (Boiy and Moens, 2009), (Kiritchenko et al., 2014), (Wagner et al., 2014) 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 ((Liu and Zhang, 2017), (Chen et al., 2017), (He et al., 2018)) 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.

Neural attention-based aspect extraction model (ABAE) is proposed in (He et al., 2017). It explicitly encodes word-occurrence statistics into word embeddings while attention is used to remove irrelevant tokens. During training ABAE uses hinge loss that maximizes the inner product between reconstructed sentence embedding and target sentence embedding and simultaneously minimizes the inner product between the reconstructed embedding and the negative sample embeddings.

SparTerm (Bai et al., 2020) consists of two parts: the first one predicts the semantic importance of each term in the vocabulary, while the second controls which terms should appear in the final sparse representation. This two models provide a term-based sparse representation based on the semantic relationship of the input text with each term in the vocabulary. Our model hugely relies on SparTerm idea complementing it with the aspect extraction model which enriches information of the summarization model.

3 Model Description

This section presents the general overview of our extractive summarizer, its architecture and the corresponding training strategy. Our model consist of two parts: summarization model and aspect embedding model (Figure 1).

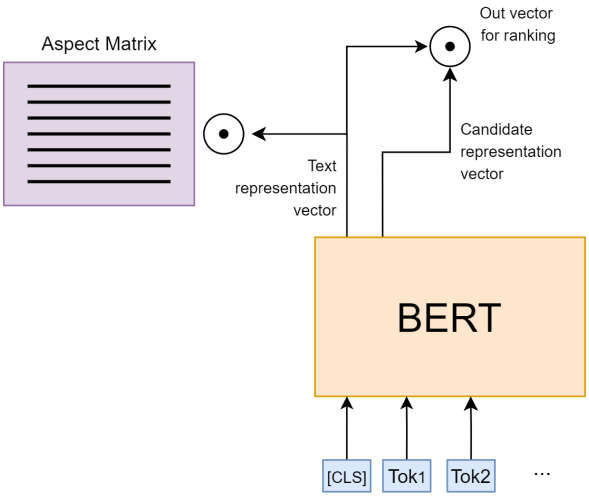

Figure 1: The proposed model: summarization module + aspect extraction module

The main idea is to explicitly use aspects of the input text to enrich the information for the summarization model.

3.1 Summarization model

Since we are doing extractive summarization, the purpose of summarization is to get a ranking for each sentence of the text.

Pretrained language models (e.g. BERT (Devlin et al., 2018)) provide contextualized representations, which makes them a good source of concise textual embeddings. As a summarization model we use a pretrained BERT with specifically designed input. During training, the input for the model is represented as a triplet of ($text$, $pos\_sentence$, $neg\_sentence$), where $text$ - is the text that needs to be summarized, $pos\_sentence$ is a sentence from summary, and $neg\_sentence$ is a sentence not from summary. The model itself is a pretrained BERT (we used *bert-base-uncased* variation from the *transformers* library (Wolf et al., 2020)), and the output is a triplet of items ($text'$,

$pos\_sentence'$, $neg\_sentence'$) which are representations of corresponding input items.

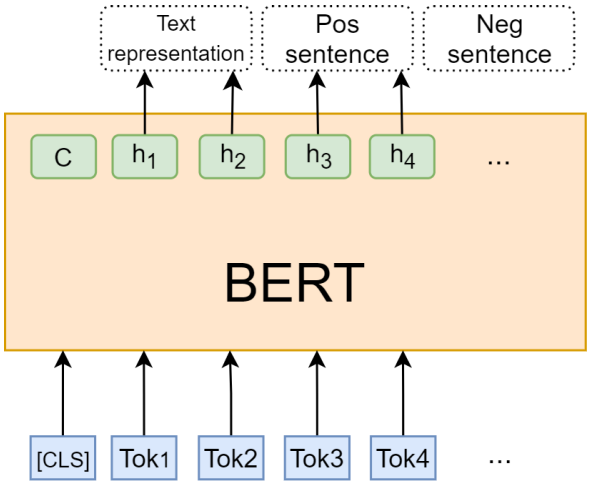

Figure 2: Summarization model

We aim to make representation of positive sentence as close as possible to representation of text and simultaneously make representation of negative sentence as far as possible, which is described in more detail in the Metrics section.

3.2 Aspect extraction model

One can get the summarization already at the previous stage by comparing the similarity metric between the text representation vector and the $candidate\_sentence$ representation vector. But quality of summary can be increased by adding information about aspects of the text as shown in the Experiments section.

The main goal of aspect extraction model is to learn a set of aspect embeddings. In general, we want the vector representation to capture the most relevant information with regards to the aspect of the input.

The model contains a matrix of aspects of size $\mathbb{R}^{K \times n}$, where K - is the number of aspects, n - embedding space size. For each input, we calculate its weights by comparing the dot product with each current aspect embedding:

$$p_i = \vec{t_i} \cdot \overrightarrow{\text{CLS}_i} \tag{1}$$

where t_i - is i-th aspect embedding in embedding matrix.

Obtained weights are then normalized with softmax function. Next, each aspect vector is multiplied by the corresponding weight and summed up to get the output reconstructed vector:

$$\overrightarrow{\text{output}} = \sum_{i=1}^{K} p_i \vec{t_i} \tag{2}$$

3.3 Training Process

Summarization model Let $R = \{(t_1, s_{1,+}, s_{1,-}), ..., (t_N, s_{N,+}, s_{N,-})\}$ denote a set of N training instances; each containing a text t_i, a positive candidate sentence $s_{i,+}$ and a negative one $p_{i,-}$, indicating that $s_{i,+}$ is more relevant to the text than $s_{i,-}$. The summarization model is trained end-to-end by optimizing the ranking objective. The loss function is the negative log likelihood of the positive sentence:

$$L_{summ}(t_i, s_{i,+}, s_{i,-}) = -\log \frac{e^{sim(t_i', s_{i,+}')}}{e^{sim(t_i', s_{i,+}')} + e^{sim(t_i', s_{i,-}')}}, \tag{3}$$

where t_i', $s_{i,+}'$, $s_{i,-}'$ is the sparse representation of t_i, $s_{i,+}$, $s_{i,-}$, sim denotes any similarity measurement (dot-product in our case).

Aspect extraction model Output reconstructed vector is needed to be similar to the input vector, so the loss is based on cosine distance:

$$L_{asp} = 1 - cosine\_similarity(\overrightarrow{\text{CLS}}, \overrightarrow{\text{output}}) \tag{4}$$

4 Dataset

CNN/Daily Mail (Nallapati et al., 2016) 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.

4.1 Converting to extractive dataset

Although CNN-DM dataset is originally designed for abstractive summarization, we modified it for extractive summarization using a special utility.

This utility reformats original abstractive dataset by determining the best extractive summary that maximizes ROUGE scores.

5 Experiments

5.1 Metrics

Also models are evaluated with full-length F1-scores of ROUGE-1, ROUGE-2, ROUGE-L (Lin, 2004). ROUGE-N is computed as follows:

$$ROUGE_N = \frac{\sum_{S \in Ref} \sum_{g_n \in S} Count_{match}(g_n)}{\sum_{S \in Ref} \sum_{g_n \in S} Count(g_n)}$$

where n stands for the length of the n-gram g_n, and $Count_{match}(g_n)$ is the maximum number of n-grams co-occurring 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.

5.2 Baselines

We compare our model with following models.

Extractive Models: **MatchSum** (Zhong et al., 2020): 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 (Xu et al., 2020): 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 are constructed based on RST trees and coreference mentions, encoded with Graph Convolutional Networks.

BerSumExt (Liu and Lapata, 2019): the model uses pretrained BERT with inserted $[CLS]$ tokens at the start of each sentence to collect features for the sentence preceding it.

Abstractive Models: **SimCLS** (Liu and Liu, 2021): 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 (Dou et al., 2021): the model has two endoders which encode the source document and guidance signal, which are attended to by the decoder.

ProphetNet (Qi et al., 2020): 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.

5.3 Experiment Setup

For the summarization model, a pre-trained BERT was used. The order of $pos\_sentence$ and $neg\_sentence$ in the input was randomly chosen to make the model actually learn the meaning of the sentences.

Then aspect model was trained with frozen BERT's weights. The achieved result was good enough after 2 epochs.

Predicted by the summarization model weights for every sentence were filtered by threshold.

For text representation, $m = 3$ of the most similar aspects vectors from the aspects matrix were calculated and averaged, the same operation for the candidate's sentence. They are also compared and filtered by similarity distance. This additional filtering improves the results, as shown below.

6 Results

We compared our model with current state of the art. We evaluate the models on the CNN/DailyMail dataset in non-anonymized version. The evaluation results are presented in Tab. 1. One could see that our model shows the superior performance among the extractive models by the means of ROUGE-2 and ROUGE-L. ROUGE-1 evaluation result for our model is 1 point 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.

In addition, we compare our model with abstractive models. The results are presented in Tab. 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. ROUGE-L is evaluated only 1 point lower that 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 aspect extraction has significant influence on the model output, leading to improvement by 5 per cent in ROUGE-1 and ROUGE-L and by 4% in ROUGE-2.

6.1 Analysis

For every input text in test set we calculated distances between $text'$ and $pos\_sentence'$ and $text'$ and $neg\_sentence'$ respectively. As shown in figure 3, the distances between initial text representation and negative sentence representations is greater than the one between the initial text and the positive sentence. Then for these values the ROC-AUC metric has been measured (figure 4).

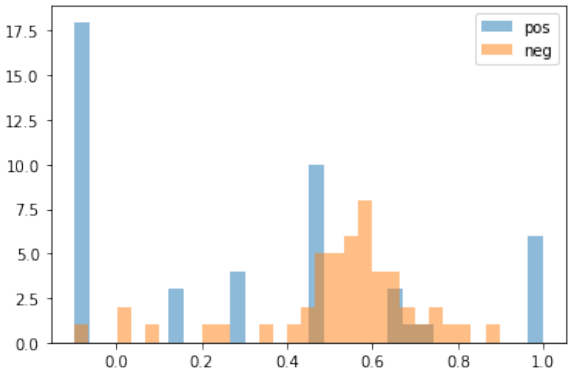

Figure 3: Comparison of distances between initial text and positive (blue) / negative (orange) sentences

7 Conclusion

We proposed new model for extractive summarization using aspects present in the input text. Our model shows state-of-the art performance on CNN/DailyMail dataset. More interestingly, we show that aspect information is crucially important for the extractive summarization. We think of that fact as a new path to follow in the extractive summarization efforts for the future research.

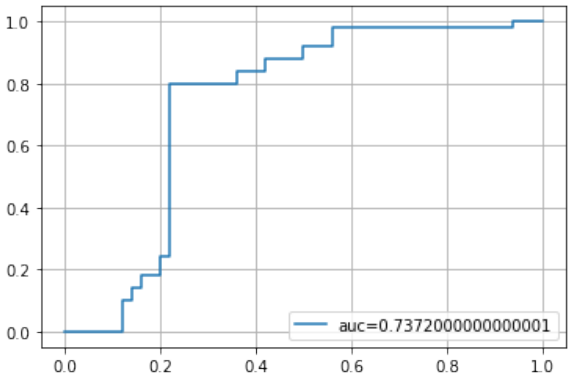

Figure 4: ROC curve over distances between initial text representation and positive/negative sentence representations

As a future work we plan to integrate aspect extraction with abstractive summarization, use other aspect extraction mechanisms.

References

Yang Bai, Xiaoguang Li, Gang Wang, Chaoliang Zhang, Lifeng Shang, Jun Xu, Zhaowei Wang, Fangshan Wang, and Qun Liu. 2020. Sparterm: Learning term-based sparse representation for fast text retrieval. *arXiv preprint arXiv:2010.00768*.

Erik Boiy and Marie-Francine Moens. 2009. A machine learning approach to sentiment analysis in multilingual web texts. *Information retrieval*, 12(5):526–558.

Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. 2017. Recurrent attention network on memory for aspect sentiment analysis. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 452–461.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Yue Dong, Andrei Romascanu, and Jackie CK Cheung. 2020. Hiporank: Incorporating hierarchical and positional information into graph-based unsupervised long document extractive summarization. *arXiv preprint arXiv:2005.00513*.

Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 2021. Gsum: A general framework for guided neural abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4830–4842.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2017. An unsupervised neural attention

Model	ROUGE-1	ROUGE-2	ROUGE-L
MatchSum (Zhong et al., 2020)	**44.41**	20.86	40.55
DiscoBERT (Xu et al., 2020)	*43.77*	20.85	*40.67*
BertSumExt (Liu and Lapata, 2019)	43.85	20.34	39.90
Ours (without aspects)	38.43	*28.51*	37.58
Ours (with aspects)	43.48	**32.52**	**42.44**

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
SimCLS (Liu and Liu, 2021)	**46.67**	22.15	**43.54**
GSum (Dou et al., 2021)	*45.94*	22.32	*42.48*
ProphetNet (Qi et al., 2020)	44.20	21.17	41.30
Ours (without aspects)	38.43	*28.51*	37.58
Ours (with aspects)	43.48	**32.52**	42.44

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 for aspect extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 388–397.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2018. Exploiting document knowledge for aspect-level sentiment classification. *arXiv preprint arXiv:1806.04346*.

Svetlana Kiritchenko, Xiaodan Zhu, Colin Cherry, and Saif Mohammad. 2014. Nrc-canada-2014: Detecting aspects and sentiment in customer reviews. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pages 437–442.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Jiangming Liu and Yue Zhang. 2017. Attention modeling for targeted sentiment. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 572–577.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740.

Yixin Liu and Pengfei Liu. 2021. Simcls: A simple framework for contrastive learning of abstractive summarization. *arXiv preprint arXiv:2106.01890*.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020. Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 2401–2410.

Joachim Wagner, Piyush Arora, Santiago Cortes, Utsab Barman, Dasha Bogdanova, Jennifer Foster, and Lamia Tounsi. 2014. Dcu: Aspect-based polarity classification for semeval task 4.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Jiacheng Xu, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Discourse-aware neural extractive text summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5021–5031.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuan-Jing Huang. 2020. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208.