Leveraging Summarization for Unsupervised Dialogue Topic Segmentation

Aleksei Artemiev[3*], **Daniil Parinov**[3*], **Alexey Grishanov**[1, 4*], **Ivan Borisov**[2], **Alexey Vasiliev**[1], **Daniil Muravetskii**[3, 4], **Aleksey Rezvykh**[2], **Aleksei Goncharov**[3, 4], and **Andrey Savchenko**[1]

[1]Sber AI Lab, Moscow, Russia
[2]Sber, Moscow, Russia
[3]MIL team, Moscow, Russia
[4]Moscow Institute of Physics and Technology, Moscow, Russia
grishanov.av@phystech.edu

Abstract

Traditional approaches to dialogue segmentation perform reasonably well on synthetic or written dialogues but suffer when dealing with spoken, noisy dialogs. In addition, such methods require careful tuning of hyperparameters. We propose to leverage a novel approach that is based on dialogue summaries. Experiments on different datasets showed that the new approach outperforms popular state-of-the-art algorithms in unsupervised topic segmentation and requires less setup.

1 Introduction

Due to online communication's growth, topic segmentation is becoming increasingly relevant (Solbiati et al., 2021). The objective of topic segmentation is "to construct a system which identifies locations in a text stream where the topic changes" (Beeferman et al., 1999). It is an example of a classic and still challenging task to automate (Park et al., 2023; Nair et al., 2023).

The challenging nature of this problem comes from several aspects. First, even for human annotators, topic segmentation might be a difficult task (Gruenstein et al., 2008), which makes unsupervised approaches preferable. Second, it is hard to handle unstructured textual datasets, especially for noisy spoken dialogues.

Driven by these challenges, we propose the use of summarization for unsupervised topic segmentation. Summary is used to extract key information from less structured dialogue data and further enhanced by Savitzky–Golay smoothing (Section 3.3) to handle high-frequency topic signals.

Using the chunking technique, we adopt this method for the limited context size of summarization models (Section 3.3). It is experimentally demonstrated that the resulting approach holds good quality for different summary models, with context sizes varying from 5 hundred to 16 thousand tokens.

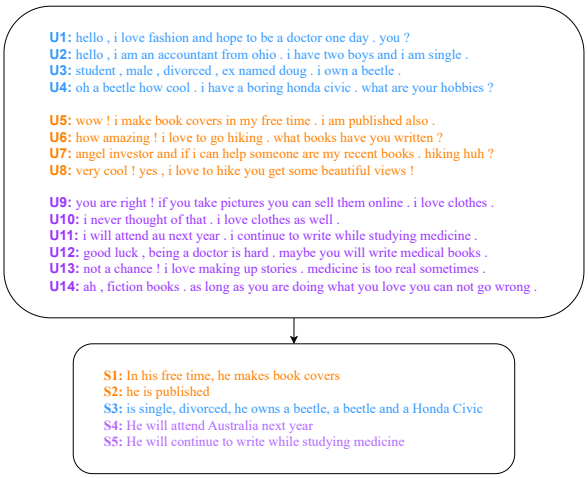

Figure 1: Reference dialogue from TIAGE dataset and simple sentences from its summary.

To the best of our knowledge, there has been no other study focusing specifically on the summary-based unsupervised topic segmentation. For a study closest to our work, (Cho et al., 2022) learned summarization and segmentation simultaneously to obtain robust sentence representations.

We have made the source code publicly available[1].

Our main contributions:

1. We leverage the summarization technique for topic segmentation of actual, noisy texts with the target domain of transcribed spoken dialogues.

2. We show that the resulting approach holds favorable quality on three datasets (SuperDialseg, TIAGE, QMSum).

[*]These authors contributed equally to this work

[1]https://github.com/milteam/unsupervised-summary-based-segmentation

Findings of the Association for Computational Linguistics: NAACL 2024, pages 4697–4704
June 16-21, 2024

3. The proposed approach also has fewer hyperparameters to tune than other unsupervised approaches.

2 Related work

2.1 Unsupervised topic segmentation

Most unsupervised topic segmentation approaches are based on the TextTiling paper (Hearst, 1997).

2.1.1 TextTiling

TextTiling can be divided into two primary components: the extraction of sentence vectors and the derivation of depth scores. While the methodology for computing depth scores remains relatively consistent or may undergo minimal modifications, calculating sentence vectors has progressed significantly from the classic Bag of Words used in TextTiling. Here, we briefly review some of the more modern approaches in historical order.

TopicTiling (topic-based sentence vectors)

In 2012, the TopicTiling was introduced (Riedl and Biemann, 2012). It is a classic approach for text segmentation that outperforms TextTiling and remains popular. Original TextTiling utilizes the Latent Dirichlet allocation (LDA) model for sentence vector calculations under the hood.

The LDA (Blei et al., 2003) is probably the most popular probabilistic topic model. It is a two-level Bayesian generative model with topic distributions over words and document distributions over topics generated from prior Dirichlet distributions.

To calculate sentence vectors, another topic model may also be used (Vorontsov et al., 2015; Tutubalina and Nikolenko, 2015, 2018). For example, *BERTopic* (Grootendorst, 2022) utilizes neural embeddings, clustering , and a class-based TF-IDF procedure to create a topic model.

Embedding-based sentence vectors

Another group of methods vectorize source text using neural embeddings from pre-trained language models and calculate the distance between adjacent pieces. Obtained distances are then employed to decide whether two adjacent sentences relate to the same segment.

BERTSeg (Solbiati et al., 2021) obtains sentence vectors from SBERT (Reimers and Gurevych, 2019) embeddings.

2.1.2 Alternative approaches

DialStart and *CohereSeg* methods (Gao et al., 2023; Xing and Carenini, 2021) utilize the Next Sentence Prediction (NSP) task from classic BERT as a scoring model to measure the coherence score (similarity) between adjacent utterances.

Recently proposed *HyperSeg* model (Park et al., 2023) leverages the probabilistic orthogonality of randomly drawn vectors at extremely high dimensions.

2.2 Supervised topic segmenation

This section briefly mentions supervised models for topic segmentation, with our primary focus on unsupervised models.

One notable supervised model, (Koshorek et al., 2018), employs a stack of two LSTM networks. The first LSTM serves as a sentence encoder, while the second classifies sentences as indicative of the beginning of a new topic.

Other approaches include hierarchical architectures. For example, *Bi-H-LSTM* (Masumura et al., 2018) introduces a hierarchical LSTM approach with additional speaker embeddings for improved segment boundary identification.

3 Method

3.1 Task formulation

Consider corpus D of documents d. Every document $d = (s_j)_{j=1}^{n}$ consists of utterances $s_1, \ldots, s_n$. This paper will use sentences as utterances if not explicitly stated. In general, they might also be replicas, words, etc.

Given document $d = (s_j)_{j=1}^{n}$ the objective of segmentation is «automatically partitioning text into coherent segments» (Beeferman et al., 1999).

3.2 Sentence vectors extraction for unstructured dialogues

The preference between spoken and written dialogues lies in their contrasting nature (Daminova, 2023; Drieman, 1962). These differences are:

1. Spoken language may contain rapidly shifting low-granularity topics.

2. Spoken language tends to be less formal and structured, often featuring repetitive and incomplete sentences.

3. Spoken language tends to be more lengthy, with more words of single syllables.

Here is our proposal to benefit in the domain mentioned above (transcribed dialogues):

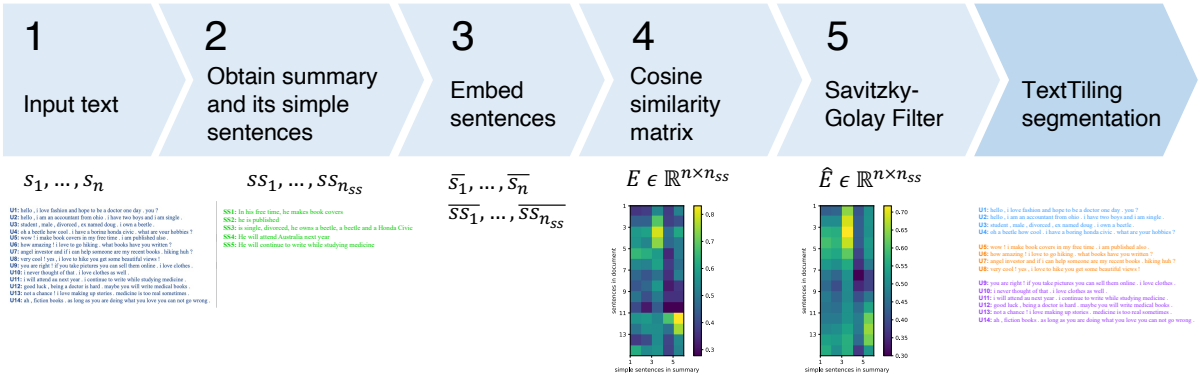

Figure 2: Unsupervised summary-based text segmentation pipeline.

1. Use the summary to obtain sentence vectors for TextTiling.

2. Use Savitzky–Golay filter (Savitzky and Golay, 1964), which is known to smooth out high-frequency noisy signals effectively (examples are available in Appendix C).

Our topic segmentation approach consists of 2 stages: proposed summary-based sentence vector extraction and a traditional segmentation scheme based on TextTiling.

3.3 Proposed summary-based sentence vectors extraction

Given document $d = (s_j)_{j=1}^{n}$, we propose to perform the following steps:

Step 1. Obtain document summary.

One may use a pre-trained model for summary extraction or use existing summaries. We describe compared pre-trained summarization models in section 4.4.

When dialogue fits the context size of the neural model, the summary is obtained for the whole dialogue. Otherwise we split a document into consecutive parts (chunks) of a size suitable for the summarization model. Then, each chunk is individually summarized, and the resulting summaries are joined together.

Step 2. Extract simple sentences (sentences that contain only one verb) $ss_1, \dots, ss_{n_{ss}}$ from the summary.

For this task, we utilized NLTK sentence parser and spaCy DependencyParser to create a grammar tree of a sentence. First, we find the root token (i.e., the main verb) and the other verbs of the sentence. Second, we find the token span for each of the other verbs. Finally, we go through all the verb's children and obtain this verb's simple sentence by the leftmost and rightmost child's indexes.

Step 3. Embed sentences $s_1, \dots, s_n$ from the source document and simple sentences $ss_1, \dots, ss_{n_{ss}}$ from the summary of the document using sentence embeddings.

Step 4. Compute cosine similarities between the embeddings of text sentences and the embeddings of simple sentences from the summary.

As a result, we obtain the matrix $E \in \mathbb{R}^{n \times n_{ss}}$ with summary-based sentence vectors.

Step 5. Apply Savitzky–Golay filter (Savitzky and Golay, 1964) to each row of $E \in \mathbb{R}^{n \times n_{ss}}$.

As a result, we obtain the matrix $\hat{E} \in \mathbb{R}^{n \times n_{ss}}$ with final summary-based sentence vectors, smoothed to handle high-frequency noisy signal.

3.4 Segmentation scheme (TextTiling)

For the rows of the matrix $\hat{E}$, the TextTiling algorithm is applied.

Consider sentence vector $(\hat{p}_j)_{j=1}^{n} = \hat{E}_j$ (row with index j in matrix $\hat{E}$). For sentence vectors, we compute the depth scores:

$$\text{depth}_j = \frac{1}{2}\left(\text{hl}_j + \text{hr}_j - 2c_j\right), \qquad (1)$$

where c_j represents the cosine similarity between left $(\hat{p}_{j-\text{window\_size}+1}, \dots, \hat{p}_j)$ and right $(\hat{p}_{j+1}, \dots, \hat{p}_{j+\text{window\_size}})$ mean-pooled windows of size *window_size*, hl_j identifies the closest local maximum on the left of index j in the similarity scores, and hr_j does the same for the right side.

The model predicts segment boundary when depth_j exceeds the threshold and c_j is the local minimum.

4 Experimental setup

4.1 Datasets

We have selected three popular dialog datasets, different in statistics domains and speech type (written/spoken):

SuperDialseg (Jiang et al., 2023) is a large-scale supervised dataset for dialogue segmentation that contains 9K dialogues based on two prevalent document-grounded dialogue corpora. The dataset was created with a feasible definition of dialogue segmentation points with the help of document-grounded dialogues, which allows for a better understanding of conversational texts.

TIAGE (Xie et al., 2021) is a dialog benchmark that considers topic shifts created through human annotations. It enables three tasks to study different scenarios of topic-shift modeling in dialog settings: detecting topic-shifts, generating responses triggered by topic-shifts, and creating topic-aware dialogs.

QMSum (Zhong et al., 2021) is designed for query-based multi-domain meeting summarisation and includes 1,808 pairs of queries and summaries from 232 meetings across various domains. The benchmark was created through human annotation of Product **AMI** (Shriberg et al., 2004), Academic **ICSI** (Shriberg et al., 2004), and Committee meetings. In QMSUM, we use the provided segmentation and treat all intermediate gaps as segments.

Dataset statistics are available in Table 1. Every dataset has pre-defined train/validation/test splitting. We use the validation set to tune hyperparameters and the test set to calculate the metrics. In the preprocessing stage, we use utterances from all speakers in a dialogue. For a summary-based pipeline, we concatenate these utterances.

4.2 Metrics

Two widely known text segmentation metrics are used: PK (Beeferman et al., 1999) and WindowDiff (WD) (Pevzner and Hearst, 2002).

Metrics description is available in Appendix A.

4.3 Baselines

Unsupervised models

First, we include two simple baselines for comparison: *random* places boundaries with a probability of the inverse average reference segment length, *absence* returns no boundaries. Despite the simplicity of these baselines, they often manage to get high segmentation metrics. For example, *random* results were mentioned even in the original SuperDialseg article (Jiang et al., 2023).

Second, we compare with the unsupervised models also extracting sentence vectors: *TT+BERTopic*, based on (Grootendorst, 2022) and *BERTSeg* (Solbiati et al., 2021).

We also included three recent state-of-the-art baselines: *DialStart* (Gao et al., 2023), *Hyperseg* (Park et al., 2023), and *CohereSeg* (Xing and Carenini, 2021).

For *CohereSeg* model, we report results with a coherence scorer based on a pre-trained BERT model (aws-ai/dse-bert-base) for a fair comparison. Full *CohereSeg* requires huge (20+ hours on A100 GPU) fine-tuning on DailyDialog pairwise samples. It would be correct to fine-tune our summary model on the equivalent dataset for a valid comparison with a fine-tuned CohereSeg.

Supervised model

Finally, we compare unsupervised approaches with the supervised *Bi-H-LSTM* model (Masumura et al., 2018).

4.4 Summary models

Our experiments compare four pre-trained summarization models for the English language (Appendix B.1).

For different languages, either a multilingual or adopted summarization model is needed, and the preprocessing steps need to be updated correspondingly.

Details about running time are available in Appendix B.2.

5 Results

5.1 Main results

In this study, we found that the proposed summary-based unsupervised method outperformed the popular unsupervised BERTSeg across all datasets and metrics (see Table 2). At best, our method surpassed BERTSeg by 5% on WD and 6% on PK. Notably, our model excelled in processing transcribed dialogues (QMSum), significantly outperforming the supervised method.

5.2 Comparison with supervised model

It is worth noting that on long documents (QMSum), supervised models Bi-H-LSTM show poor quality due to the training data's small amount

Table 1: **Statistics of datasets**

Dataset	# docs			# words in doc			avg #		
	train	val	test	min	avg	max	words in section	utterances in doc	utterances in section
Super-DialSeg	6690	1298	1277	33	218.3	525	48.8	13.4	3.4
TIAGE	286	96	97	109	185.1	264	40.4	15.4	4.1
QMSum	162	35	35	1371	9521.4	25529	1593.6	334.7	76.5

Table 2: **Overall performance comparison. The down arrow shows that the lower the metric value, the better. The best result is highlighted in bold, the second is underlined. An asterisk denotes a supervised model if it outperformed all unsupervised models. Bi-H-LSTM is placed separately since it is the only supervised method here.**

Models \ Datasets	SuperDialSeg		TIAGE		QMSum	
	WD↓	PK↓	WD↓	PK↓	WD↓	PK↓
Bi-H-LSTM	*0,220	*0.210	0.492	0,442	0,714	0,648
random	0.554	0.474	0.591	0.499	0.530	0.470
absence	0.533	0.533	0.520	0.520	0.404	0.404
BERTSeg	0.483	0.476	0.470	0.439	0.387	0.377
TT+BERTopic	0.489	0.478	0.478	0.461	0.447	0.438
DialSTART	0.498	0.483	0.507	0.471	0.478	0.443
HyperSeg	0.512	0.503	0.522	0.519	0.485	0.461
CohereSeg	0.562	**0.438**	0.528	0.451	0.817	0.569
SumSeg (ours)	**0.480**	0.469	**0.455**	**0.438**	**0.379**	**0.357**

Table 3: **Performance comparison of various summary models. All the summary models used chunking 3.3 on the QMSUM dataset (average dialogue length of 10k words and maximum of 25k words). The down arrow shows that the lower the metric value, the better.**

Datasets \ Models		Summary Segmentation			
		BART	BART-samsum	FLAN-T5-samsum	LED-samsum
Super DialSeg	**WD↓**	0.488	**0.480**	0.485	0.491
	PK↓	0.480	**0.469**	0.475	0.483
TIAGE	**WD↓**	**0.443**	0.455	**0.443**	0.493
	PK↓	0.415	0.438	**0.402**	0.479
QMSum	**WD↓**	0.431	**0.379**	0.410	0.436
	PK↓	0.414	**0.357**	0.399	0.419

and high diversity. In contrast, the summarization model produces good metrics.

5.3 Comparison of different summary models

In the next experiment, we assess the stability of our setup on various summarization models.

The results (Table 3) indicate that summarization models, even those not explicitly designed for dialogue summarization, effectively identify text boundaries.

For example, on the TIAGE dataset, BART achieves parity with FLAN-T5-samsum in the WD metric and is within a 3% difference in the PK metric compared to FLAN-T5-samsum.

6 Conclusion and future work

We have presented a novel summary-based approach for topic segmentation focusing on transcribed spoken dialogues. We leverage summarization for sentence vector extraction and combine it with the Savitskiy-Golay filtering (Savitzky and Golay, 1964) to handle the noisy nature of transcribed spoken data.

Experiments on three real-world datasets demonstrate the effectiveness of the proposed model among the tested unsupervised approaches.

We hope that our work can inspire further development of summary-based topic segmentation. More research steps are planned for summarization (including applying LLMs) and its use for text segmentation.

Limitations

In contrast to existing topic segmentation techniques, such as sentence embeddings, the proposed approach requires additional summarization steps, which may be time-consuming, especially for substantial data, such as wiki727. Furthermore, obtaining the pre-trained summarization model for low-resource languages might be difficult.

Ethics Statement

All the data we used in our work was anonymized. The personal information of dialogue participants was not considered and was not used for modeling or other purposes.

Acknowledgements

We would like to express our gratitude to Elena Tutubalina for numerous discussions that helped to improve our work, to Gleb Gusev for valuable comments, and to Alexandra Laricheva for stylistic help.

References

Doug Beeferman, Adam L. Berger, and John D. Lafferty. 1999. Statistical models for text segmentation. *Machine Learning*, 34(1-3):177–210.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3:993–1022.

Sangwoo Cho, Kaiqiang Song, Xiaoyang Wang, Fei Liu, and Dong Yu. 2022. Toward unifying text segmentation and long document summarization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 106–118, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

K.R. Daminova. 2023. Difference between written and spoken language. *Journal of new century innovations*, 30:66–68.

G.H.J. Drieman. 1962. Differences between written and spoken language: An exploratory study. *Acta Psychologica*, 20:36–57.

Haoyu Gao, Rui Wang, Ting-En Lin, Yuchuan Wu, Min Yang, Fei Huang, and Yongbin Li. 2023. Unsupervised dialogue topic segmentation with topic-aware utterance representation. *arXiv preprint arXiv:2305.02747*.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China. Association for Computational Linguistics.

Maarten Grootendorst. 2022. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. *arXiv preprint arXiv:2203.05794*.

Alexander Gruenstein, John Niekrasz, and Matthew Purver. 2008. Meeting structure annotation: Annotations collected with a general purpose toolkit. *Recent Trends in Discourse and Dialogue*, pages 247–274.

Marti A. Hearst. 1997. Text tiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64.

Junfeng Jiang, Chengzhang Dong, Akiko Aizawa, and Sadao Kurohashi. 2023. SuperDialseg: A large-scale dataset for supervised dialogue segmentation. *arXiv preprint arXiv:2305.08371*.

Omri Koshorek, Adir Cohen, Noam Mor, Michael Rotman, and Jonathan Berant. 2018. Text segmentation as a supervised learning task. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 469–473, New Orleans, Louisiana. Association for Computational Linguistics.

Ryo Masumura, Setsuo Yamada, Tomohiro Tanaka, Atsushi Ando, Hosana Kamiyama, and Yushi Aono. 2018. Online call scene segmentation of contact center dialogues based on role aware hierarchical LSTM-RNNs. *Proceedings of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 811–815.

Inderjeet Nair, Aparna Garimella, Balaji Vasan Srinivasan, Natwar Modani, Niyati Chhaya, Srikrishna Karanam, and Sumit Shekhar. 2023. A neural CRF-based hierarchical approach for linear text segmentation. In *Findings of the Association for Computational Linguistics (EACL)*, pages 883–893, Dubrovnik, Croatia. Association for Computational Linguistics.

Seongmin Park, Jinkyu Seo, and Jihwa Lee. 2023. Unsupervised dialogue topic segmentation in hyperdimensional space. pages 730–734.

Lev Pevzner and Marti A. Hearst. 2002. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, 28(1):19–36.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Martin Riedl and Chris Biemann. 2012. TopicTiling: A text segmentation algorithm based on LDA. In *Proceedings of ACL 2012 Student Research Workshop*, pages 37–42, Jeju Island, Korea. Association for Computational Linguistics.

Abraham. Savitzky and M. J. E. Golay. 1964. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8):1627–1639.

Elizabeth Shriberg, Raj Dhillon, Sonali Bhagat, Jeremy Ang, and Hannah Carvey. 2004. The ICSI meeting recorder dialog act (MRDA) corpus. In *Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue at HLT-NAACL 2004*, pages 97–100, Cambridge, Massachusetts, USA. Association for Computational Linguistics.

Alessandro Solbiati, Kevin Heffernan, Georgios Damaskinos, Shivani Poddar, Shubham Modi, and Jacques Cali. 2021. Unsupervised topic segmentation of meetings with BERT embeddings. *arXiv preprint arXiv:2106.12978*.

Elena Tutubalina and Sergey Nikolenko. 2015. Inferring sentiment-based priors in topic models. In *Proceedings of the Mexican International Conference on Artificial Intelligence*, pages 92–104. Springer.

Elena Tutubalina and Sergey Nikolenko. 2018. Exploring convolutional neural networks and topic models for user profiling from drug reviews. *Multimedia Tools and Applications*, 77:4791–4809.

Konstantin Vorontsov, Oleksandr Frei, Murat Apishev, Peter Romov, and Marina Dudarenko. 2015. Bigartm: Open source library for regularized multimodal topic modeling of large collections. pages 370–381.

Huiyuan Xie, Zhenghao Liu, Chenyan Xiong, Zhiyuan Liu, and Ann Copestake. 2021. TIAGE: A benchmark for topic-shift aware dialog modeling. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1684–1690, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Linzi Xing and Giuseppe Carenini. 2021. Improving unsupervised dialogue topic segmentation with utterance-pair coherence scoring. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 167–177, Singapore and Online. Association for Computational Linguistics.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, et al. 2021. QMSum: A new benchmark for query-based multi-domain meeting summarization. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (ACL): Human Language Technologies*, pages 5905–5921.

A Metrics

Pk is calculated by passing a sliding window of length k through the document's text. The k value is half the reference segment's average length.

$$k = \frac{N}{2 * number\ of\ bounderies}$$

Where N is the total number of sentences (or content utterances).

At each iteration, the algorithm determines whether the two ends of the frame are in the same or different segments of the reference segmentation and increases the counter if the model's segmentation does not agree with the reference one.

The number of measurements normalizes the resulting value to get a value from 0 to 1.

WindowDiff is obtained by summing the differences of the ends of the segments in the reference segmentation $R_{i,i+k}$ and in the computed segmentation made by model $C_{i,i+k}$. Suppose it is greater than zero (i.e., the number of segments in the reference segmentation differs from the segmentation made by the model). In that case, it is summed with the rest and then also normalized by the total number of measurements:

$$WindowDiff = \frac{1}{N-k} \sum_{i=1}^{N-k} [R_{i,i+k} \neq C_{i,i+k}],$$

where k and N are defined similarly to the previous paragraph.

B Implementation details

B.1 Summarization models used

For comparison, we select four popular open-source models for abstractive summarization from HuggingFace with different context sizes.

A list of models is:

1. **BART:** facebook/bart-large-cnn, context size is 1024
2. **BART-samsum:** philschmid/bart-large-cnn-samsum, context size is 1024
3. **FLAN-T5:** philschmid/flan-t5-base-samsum, context size is 512
4. **LED:** rooftopcoder/led-base-book-summary-samsum, context size is 16384

Some models have the suffix 'samsum', meaning that a model was fine-tuned using the SAMSum corpus (Gliwa et al., 2019), collected from manually annotated summaries for chat dialogues.

B.2 Computational time

It takes approximately two hours to pick up parameters on three datasets for one summarization model.

Inference time for summarization models is available in Table 4.

Table 4: **Model inference time**

Model	Inference time, sec
BART	7.5
BART-samsum	6.6
FLAN-T5-samsum	19.2
LED-samsum	0.8

C Savitzky–Golay smoothing examples

Smoothing out sentence vectors with Savitzky–Golay filter often helps to clarify segmentation boundaries. Below, we give two examples from datasets used in experiments.

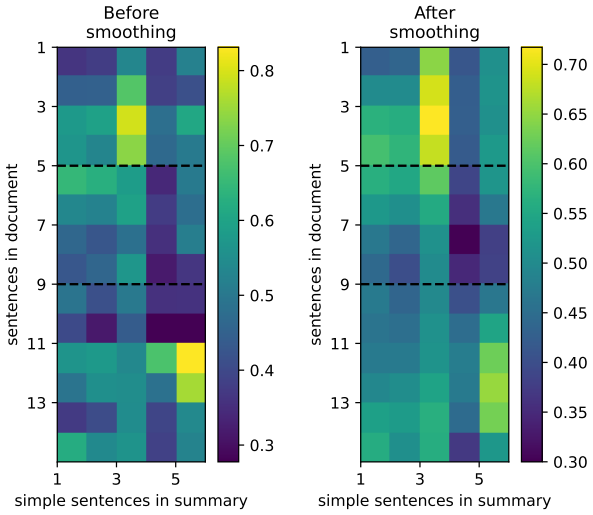

Figure 3: The effect of Savitzky–Golay smoothing. Sentence vectors of TIAGE document example. Dotted lines indicate segment boundaries.

Heat maps (Figures 3, 4) are colored based on the cosine distance between the embeddings of document sentences and the simple sentences of the summary, before and after applying the Savitzky–Golay filtering.

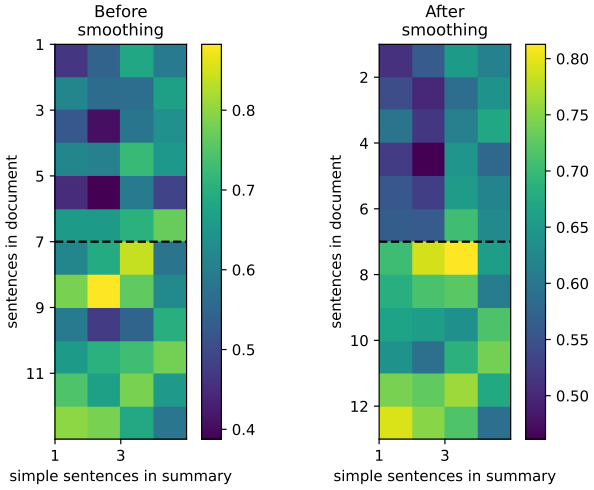

Figure 4: The effect of Savitzky–Golay smoothing. Sentence vectors of SuperDialseg document example. The dotted line indicates the segment boundary.

It can be seen that adding a filter makes segment boundaries more obvious.