Leveraging summarization for unsupervised topic segmentation of long dialogues

Anonymous NAACL-HLT 2024 submission

Abstract

Traditional approaches to dialogue segmentation perform quite well on synthetic or short dialogues but suffer when dealing with long, 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 SotA algorithms in unsupervised topic segmentation and requires less setup. The source code is available at https://anonymous.4open.science/r/unsupervised-summary-based-segmentation

1 Introduction

The objective of topic segmentation is "to construct a system which, when given a stream of text, identifies locations where the topic changes" (Beeferman et al., 1999). This is an example of a classic and still challenging task to automate (Bai et al., 2023), (Nair et al., 2023).

The challenging nature of topic segmentation comes from several aspects. First, even for human annotators topic segmentation might be a hard task (Gruenstein et al., 2008), which makes unsupervised approaches preferable. Second, it is hard to handle unstructured textual datasets, especially for long noisy real dialogues (section 3.2).

Driven by these challenges, we propose the use of summary for unsupervised topic segmentation We also adopt this method for the limited context size of summarization models by using the chunking technique (section 1). The resulting approach holds good quality for different models, with context size from 512 to 16384 tokens (table 3).

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.

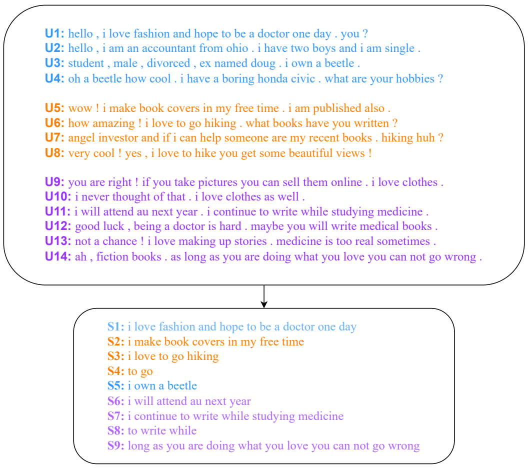

Figure 1: Reference dialogue and generated summary. Example from TIAGE dataset.

Our main contributions:

1. We leverage the summarization technique for topic segmentation of long noisy texts, especially from transcribed spoken dialogues.

2. We show that the resulting approach holds better quality on 3 datasets (SuperDialseg, TIAGE, QMSum).

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

2 Related work

Most approaches are to unsupervised topic segmentation based on TextTiling work (Hearst, 1997).

2.1 TextTiling

TextTiling can be divided into two primary components: the computation 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.

2.1.1 TopicTiling

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

Latent Dirichlet allocation (LDA) (Blei et al., 2001) is the most popular probabilistic topic model. LDA is a two-level Bayesian generative model, in which topic distributions over words and document distributions over topics are generated from prior Dirichlet distributions.

To calculate topic vectors, other topic model may also be used. *BERTopic* (Grootendorst, 2022) utilizes neural embeddings, clustering, and class-based TF-IDF procedure to create a topic model.

2.1.2 Embedding-based topic segmentation

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) utilizes SBERT (Reimers and Gurevych, 2019) embeddings to segment dialogue utterances.

Some other 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.

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

3 Method

3.1 Task formulation

Consider corpus D of documents d. Every document $d = (s_j)_{j=1}^n$, consists of utterances $s_1, \dots, s_n$. In this paper, we 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 goal of segmentation is to find a partition $L = (l_j)_{j=1}^k$ such that joining the elements (segments) of L in the same order reconstructs d and $l_i \cap l_j = \emptyset \quad \forall i \neq j$.

3.2 Handling unstructured dialogues

We propose to narrow focus on transcribed spoken dialogues. The preference between spoken and written dialogues lays in their contrasting nature(Daminova, 2023), (Drieman, 1962):

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.

3.3 Proposed summary-based pipeline

Given document $d = (s_j)_{j=1}^n$:

1. Obtain document summary using a neural network model. When dialogue fits the context size of the 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 was individually summarized, and finally, the resulting summaries were joined together.

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, obtain this verb's simple sentence by leftmost and rightmost child's indexes.

3. Map 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 to embeddings.

4. Compute cosine proximity between embeddings of text sentences and embeddings of simple sentences from the summary. As a result, we get a matrix $E \in \mathbb{R}^{n \times n_{ss}}$

Table 1: **Statistics of datasets**

Dataset	# docs			# words in doc			avg #		
	train	val	test	min	avg	max	words in section	uttrances in doc	utterances in section
Super-DialSeg	6690	1298	1277	33.0	218.3	525.0	48.8	13.4	3.4
TIAGE	286	96	97	109.0	185.1	264.0	40.4	15.4	4.1
QMSum	162	35	35	1371.0	9521.4	25529.0	1593.6	334.7	76.5

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

6. Apply TextTiling algorithm on the rows of the matrix $\hat{E}$.

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

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

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

   For each sentence from source document s_j where depth_j exceeding the threshold and c_j is local minimum we make a decision about the presence of a segment boundary.

To benefit in aforementioned domain we propose

1. The use of summary to obtain sentence vectors for TextTiling (stages 1-4).

2. Use Savitzky–Golay filter (Savitzky and Golay, 1964) (stage 5). This filter is known to effectively smooth out high-frequency noisy signals.

4 Experimental setup

4.1 Datasets

We have selected 3 popular dialog datasets.

In the preprocessing stage, we use utterances from all of the speakers in a dialogue. For a summary-based pipeline, we concatenate these utterances.

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.

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 is 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 benchmark (Zhong et al., 2021) is designed for the task of 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.

4.2 Metrics

Two widely known text segmentation metrics are used: PK (Beeferman et al., 1999) and WindowDiff (WD) (Pevzner and Hearst, 2002). Their detailed description is available at Appendix A.

4.3 Models

We compare the proposed approach with the unsupervised models from section 2: *TT+BERTopic*, *BERTSeg* (Solbiati et al., 2021), *DialStart* (Gao et al., 2023), *CohereSeg* (Xing and Carenini, 2021), and *Hyperseg* (Park et al., 2023).

We also included two baselines for comparison: *random* places boundaries with a probability of the inverse average reference segment length, *absence* returns no boundaries.

For a fair comparison, we report CohereSeg results with a coherence scorer based on a pre-trained BERT model (aws-ai/dse-bert-base). Full CohereSeg requires huge (20+ hours on A100 GPU) fine-

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+BERTTopic	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
BART-samsum (ours)	**0.480**	0.469	**0.455**	**0.438**	**0.379**	**0.357**

Table 3: **Performance Comparison of different summary models. All of the summary models used chunking 1 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

tuning on DailyDialog pairwise samples. This will increase TIAGE's metrics to a new top-1. For a valid comparison with a fine-tuned CohereSeg, it would be correct to also fine-tune our summary model on the equivalent dataset.

5 Experimental results

5.1 Main results

In our study, we found that our 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), it significantly outperformed the supervised method.

5.2 Comparison of different summary models.

We assess the stability of our setup using various summarization models, as detailed in Table 3.

The results indicate that summarization models, even those not specifically designed for dialogue summarization, are effective in using for identifying 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 when compared to FLAN-T5-samsum.

6 Conclusion and future work

We have presented a novel approach for topic segmentation based on summary.

We give practical evidence that the proposed approach shows favorable performance among the tested unsupervised approaches and theoretical evidence that the proposed summary-based method is especially suitable for the transcribed spoken dialogues domain.

We hope that our work can inspire further development of summary-based topic segmentation.

Further research steps are planned for summarization and its use for text segmentation.

Limitations

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

Ethics Statement

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

Acknowledgements

We thank anonymous reviewers for their fruitful comments and feedback.

References

Haitao Bai, Pinghui Wang, Ruofei Zhang, and Zhou Su. 2023. Segformer: A topic segmentation model with controllable range of attention. pages 12545–12552. AAAI Press.

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

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2001. Latent dirichlet allocation. In *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada]*, pages 601–608. MIT Press.

Sangwoo Cho, Kaiqiang Song, Xiaoyang Wang, Fei Liu, and Dong Yu. 2022. Toward unifying text segmentation and long document summarization. In *Proceedings of the 2022 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.

Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure.

Alexander Gruenstein, John Niekrasz, and Matthew Purver. 2008. *Meeting Structure Annotation*, 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.

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 2023*, 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 2019 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. *Anal Chem*, 36(8):1627–1639.

Alessandro Solbiati, Kevin Heffernan, Georgios Damaskinos, Shivani Poddar, Shubham Modi, and Jacques Cali. 2021. Unsupervised topic segmentation of meetings with bert embeddings.

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 Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. 2021. Qmsum: A new benchmark for query-based multi-domain meeting summarization.

A Metrics

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

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

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 segmentation of the model does not agree with the reference one.

The resulting value is normalized by the number of measurements to get a value in the range 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 $C_{i,i+k}$. If it is greater than zero (i.e., the number of segments in the reference segmentation differs from the segmentation made by the model), it is summed with the rest, and then also normalized by the total number of measurements:

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

k, N defined similarly to the previous paragraph

B Implementation details

B.1 Computational time

It takes roughly two hours to pick up parameters on 3 datasets for one summarization model. Model inference time represents 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

B.2 Summarization models used

For the purpose of comprehensive comparison, we select the most popular open-source models for abstractive summarization from HuggingFace.

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 of the models have the suffix 'samsum' meaning that a model was fine-tuned using the SAMSum corpus, which renders it an appropriate selection for abstractive dialogue summarization.