Leveraging summarization for unsupervised topic segmentation of long dialogues

Anonymous EACL 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.

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 according to (Gruenstein et al., 2008). Hence collecting labeled data for segmented meetings is complex and expensive and there is a lack of ground truth labeling data. Second, it is hard to handle unstructured textual datasets, especially for long noisy real dialogues.

In this work, we propose the use of summarization to handle the structure of long noisy dialogues. In the case of dialogues that exceed the context size of the model, we adopted a solution by splitting them into smaller chunks. Each chunk was individually summarized, and then the resulting summaries were joined together.

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

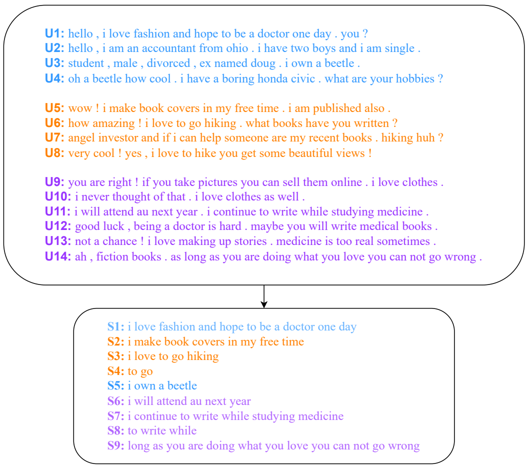

Figure 1: Reference dialogue and generated summary. Example from TIAGE dataset.

Our main contributions:

1. We leverage the summarization technique for topic segmentation of long documents.
2. We show that the resulting approach holds better quality on 3 datasets (SuperDialseg, QMSum, TIAGE).
3. The Proposed approach also has fewer hyperparameters to tune than other unsupervised approaches.

2 Related work

2.1 Unsupervised topic segmentation

Most of the existing approaches here are based on classical work TopicTiling (Riedl and Biemann, 2012).

The TopicTiling algorithm can be divided into two primary components: the computation of topic vectors and the derivation of depth scores. While the methodology for computing depth scores remains relatively consistent or may undergo minimal modifications, the process of calculating topic vectors offers different approaches. Here we briefly review some of them in historical order.

2.1.1 Topic modeling-based segmentation

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.

Later, Additive Regularization of Topic Models (ARTM) (Vorontsov et al., 2015) was introduced. The additive Regularization approach enables us to combine probabilistic assumptions with linguistic and problem-specific requirements in a single multi-objective topic model.

On the different side from probabilistic topic models such as ARTM and LDA stays BERTopic model. BERTopic generates document embedding with pre-trained transformer-based language models, clusters these embeddings, and finally, generates topic representations with the class-based TF-IDF procedure. BERTopic generates coherent topics and remains competitive across a variety of benchmarks involving classical models and those that follow the more recent clustering approach of topic modeling.

2.1.2 Embedding-based topic segmentation

Another group of methods aims to vectorize source text and calculate the distance between adjacent pieces.

Obtained distances are then employed to decide whether two neighboring sentences relate to the same topic. (Solbiati et al., 2021) utilizes siamese networks to derive semantically meaningful sentence BERT (SBERT) embeddings (insert citation here) to segment dialogue utterances. It first pretrains the encoder model on the Next Sentence Prediction (NSP) task, then uses Bert as a scoring model to measure the coherence score between adjacent utterances.

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 or not.

Other approaches include hierarchical architectures. For example, (Takanobu et al., 2018) uses a hierarchical LSTM for weakly supervised learning of token segmentation in goal-oriented dialogues. Another work, (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 and vocabulary W of all possible terms w. Every document $d = (s_j)_{j=1}^{n_d}$, consists of utterances $s_1, \ldots, s_{n_d}$ which are typically sentences (it might also be replicas or words in some topic segmentation problems).

Given document $d = (s_j)_{j=1}^{n_d}$ the goal of segmentation is to find a partition $L = (l_j)_{j=1}^{k_d}$ such that joining the elements (segments) of L in the same order reconstructs d and $l_i \cap l_j = \emptyset \quad \forall i \neq j$. Each segment $l_i \in L$ represents some topic.

3.2 TopicTiling-like pipeline for topic segmentation

Traditional topic modeling-based segmentation pipeline consists of multiple steps:

1. Construct a topic model for all corpus:

$$p(w \mid d) = \sum_{t \in T} p(w \mid t) p(t \mid d),$$

   where $d \in D, w \in W$. In the original TopicTiling LDA was used, other topic models may also be chosen, for example, BERTopic or BigARTM.

2. For particular document $d = (s_j)_{j=1}^{n_d}$ obtain topic distribution for sentence s_j:

$$p(t \mid d, s_j) = \frac{1}{|s_j|} \sum_{w \in s_j} p(t \mid d, w)$$

   and topic vector of sentence s_j:

$$p_j = \left(p\left(t \mid d, s_j\right)\right)_{t \in T}$$

3. Apply Savitzky–Golay filter (Savitzky and Golay, 1964) to p_j to get $\hat{p_j}$.

4. Run TopicTiling algorithm (Riedl and Biemann, 2012) on to the smoothed topic vectors.

   Compute depth score d_j and return candidates with d_j exceeding the threshold.

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

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

Where c_j represents the cosine similarity between left ($s_{p-\text{window}+1}, \ldots, s_p$) and right ($s_{p+1}, \ldots, s_{p+\text{window}}$) mean-pooled windows.

$hl(c_j)$ identifies the closest local maxima on the left of index j in the similarity scores.

$hr(c_j)$ does the same for the right side.

3.3 Proposed summary-based pipeline

Our proposed pipeline:

1. Document summarization using a neural network model.

2. Divide the summary of a document into simple sentences using NLTK sentence tokenizer and spacy syntax parser for tree creation. The purpose is to address only one specific topic within the document.

3. Calculate embeddings for simple sentences from the summary of the document, as well as for sentences from the source document.

4. Calculate cosine proximity between embeddings of text sentences and embeddings of simple sentences (ss) from the summary. As a result, we get a matrix $E \in \mathbb{R}^{n \times ss}$, where n is the number of sentences in the original document, ss is the number of simple sentences in the summary of the document. Similar to topic models, we call these vectors topic vectors.

5. Smoothing along initial sentences from document(in n dimension). This process is particularly advantageous for sentences devoid of topical information, a common occurrence in dialogues where the inclusion of such sentences contributes to speech fluidity and the style of the speaker.

6. Apply TopicTilling algorithm.

3.4 Comparing different summary models

We test stability of our setup with different summary models.

The key difference for our dataset choice is in input sequence length, which leads to the problem of long text chunking. The next notable difference between the models is in the time it takes them to handle long texts. For example, LED is faster than all the above models due to the large input context (16384 tokens), which allows not to divide the text into many small chunks. Based on Table 4, FLAN-T5's inference time takes the longest, BART is the trade-off in runtime between LED and FLAN-T5.

4 Experiments

We have selected 3 most popular and high-quality datasets for dialog topic segmentation. All of them are different in structure and meaning, allowing the most complete comparison of all our models.

4.1 Datasets

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.

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.

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.

Table 2: **Overall Performance Comparison. The down arrow shows that the lower the metric value, the better, the up arrow, vice versa. The best result is highlighted in bold, the second is underlined. An asterisk denotes a supervised model if it outperformed all unsupervised models.**

Datasets	Models	Unsupervised					Supervised
		Without any annotated corpus			TT+Summary	With topic modeling	
		Random	Absence	TT+SBERT	BART-samsum (our)	TT+BERTTopic	Bi-H-LSTM
Super-DialSeg	WD↓	0,554	0,533	0,483	**0,480**	0,489	*0,220
	PK↓	0,474	0,533	0,476	**0,469**	0,478	*0,210
	F1↑	**0,269**	0,000	0,127	0,170	0,138	*0,840
	Score↑	**0,378**	0,234	0,324	0,348	0,328	*0,813
TIAGE	WD↓	0,591	0,520	0,470	**0,455**	0,478	0,492
	PK↓	0,499	0,520	0,439	**0,438**	0,461	0,442
	F1↑	**0,175**	0,000	0,120	0,141	0,109	*0,430
	Score↑	0,315	0,240	0,333	**0,348**	0,320	*0,482
QMSum	WD↓	0,530	0,404	0,387	**0,379**	0,447	0,714
	PK↓	0,470	0,404	0,377	**0,357**	0,438	0,648
	F1↑	0,015	0,000	0,008	**0,017**	0,008	*0,090
	Score↑	0,258	0,298	0,313	**0,325**	0,283	0,205

4.2 Metrics

In this paper, several metrics widely known in the literature are used: PK (P_k) (Beeferman et al., 1999) and WD (WindowDiff) (Pevzner and Hearst, 2002) – metrics that use a sliding window to calculate correctly predicted boundaries. For a more convenient comparison, we use the aggregate metric *Score* proposed in (Jiang et al., 2023).

A detailed description of all metrics is presented in Appendix A.

4.3 Models

Baselines

There are 2 baselines included for comparison. Random baseline places boundaries with a probability of the inverse average reference segment length. Absence returns no boundaries. Even though they are simple, on the SuperDialseg dataset Random baseline gets a high score, which was mentioned even in the original article (Jiang et al., 2023).

Unsupervised models

For unsupervised models comparison we include BERTopic-based unsupervised model as defined in 3.2 and (Solbiati et al., 2021) close to state-of-the-art.

Supervised models

Finally, we compare against the bidirectional H-LSTM supervised model based on (Masumura et al., 2018).

5 Results and analysis

As shown in Tables 2 and 3, our unsupervised method based on using TopicTiling model with summary-based topic vectors obtains better results on each dataset and metrics than the most popular SotA approaches in unsupervised topic segmentation – TopicTiling over BERT embeddings. It is worth noting that on long documents (QMSum) supervised models show poor quality, while the summarization model on the contrary shows good metrics. At best, our algorithm outperforms TopicTiling over BERT embeddings by 5% on WD, 6% on PK, 114% on F1, and 21% on total score.

6 Conclusion and future work

We have presented and investigated a novel approach to segment dialog data using summarization models, which shows better metrics among the tested unsupervised approaches. The BART-samsum model showed the best results; it outperforms other unsupervised models not only in metrics but also in ease of configuration. Although on some datasets summary-based models are inferior to the supervised approach, they nevertheless deserve a lot of attention because do not require careful marking.

Further research steps are planned to investigate the application of LLM to text segmentation and summarization and the use of this information for 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 all the 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.

Alexander Gruenstein, John Niekrasz, and Matthew Purver. 2008. *Meeting Structure Annotation*, pages 247–274.

Junfeng Jiang, Chengzhang Dong, Akiko Aizawa, and Sadao Kurohashi. 2023. Superdialseg: A large-scale dataset for supervised dialogue segmentation.

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. *2018 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 2023*, pages 883–893, Dubrovnik, Croatia. Association for Computational Linguistics.

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

Martin Riedl and Chris Biemann. 2012. Topictiling: A text segmentation algorithm based on lda. In *Annual Meeting of the 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.

Ryuichi Takanobu, Minlie Huang, Zhongzhou Zhao, Feng-Lin Li, Feng Ji, Haiqing Chen, Xiaoyan Zhu, and Liqiang Nie. 2018. A weakly supervised method for topic segmentation and labeling in goal-oriented dialogues via reinforcement learning. In *International Joint Conference on Artificial Intelligence*.

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.

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

Table 3: **Performance Comparison of different summary models. The down arrow shows that the lower the metric value, the better, the up arrow, vice versa.**

Datasets \ Models		TT+Summary			
		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
	F1↑	0,136	**0,170**	0,143	0,154
	Score↑	0,326	**0,348**	0,331	0,334
TIAGE	**WD↓**	**0,443**	0,455	**0,443**	0,493
	PK↓	0,415	0,438	**0,402**	0,479
	F1↑	**0,234**	0,141	0,177	0,097
	Score↑	**0,403**	0,348	0,377	0,305
QMSum	**WD↓**	0,431	**0,379**	0,410	0,436
	PK↓	0,414	**0,357**	0,399	0,419
	F1↑	**0,019**	0,017	0,000	0,008
	Score↑	0,298	**0,325**	0,298	0,290

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

F1 (f1-score) is a classical metric that uses boundaries as classes in a binary classification problem. In this setting, class 1 means the beginning of a new segment, and 0 means the continuation of the section. The metric is calculated using the following formula:

$$F_1 = \frac{2 * precision * recall}{precision + recall}$$

Score is the aggregation of the three previous metrics.

$$Score = \frac{2 * F1 + (1 - P_k) + (1 - WD)}{4}$$

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

B.2 Summarization models used

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

A list of models is:

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

   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.

C Comparing different summarization models

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