BP-Seg: A graphical model approach to unsupervised and non-contiguous text segmentation using belief propagation

Fengyi Li, Kayhan Behdin, Natesh Pillai, Xiaofeng Wang, Zhipeng Wang, Ercan Yildiz
LinkedIn Corporation
{fenli, kbehdin, napillai, xiaofwan, zhipwang, eyildiz}@linkedin.com

Abstract

Text segmentation based on the semantic meaning of sentences is a fundamental task with broad utility in many downstream applications. In this paper, we propose a graphical model-based unsupervised learning approach, named BP-Seg for efficient text segmentation. Our method not only considers local coherence, capturing the intuition that adjacent sentences are often more related, but also effectively groups sentences that are distant in the text yet semantically similar. This is achieved through belief propagation on the carefully constructed graphical models. Experimental results on both an illustrative example and a dataset with long-form documents demonstrate that our method performs favorably compared to competing approaches.

1 Introduction

Segmenting text into semantically coherent segments has been a long-studied problem in the field of natural language processing [18, 1]. The applications of text segmentation range from information retrieval [27, 10], document summarization [4, 16], disclosure analysis [24, 22], and optimizing prompts for large language models (LLMs) by extracting the most relevant parts [14, 25].

Traditional methods, whether supervised [12, 1, 7] or unsupervised [9, 8, 2], mainly focus on *contiguous* or *sequential* text segmentation. The goal is to cluster *consecutive* sentences in a way that ensures those within the same group are semantically more similar to each other than to sentences in different groups. For example, if a text consists of five sentences labeled $\{1, 2, 3, 4, 5\}$, traditional methods might segment them into groups such as $\{1, 2\}$ and $\{3, 4, 5\}$. However, in practice, it is sometimes the case that $\{1, 5\}$ are more semantically similar and should form one group, while $\{2, 3, 4\}$ form another. On the other hand, some frameworks for text segmentation disregard the adjacency relationships between sentences and the overall structure of the text. For example, methods such as k-means [15] treat each sentence (or its embedding) as an isolated data instance, without considering that a sentence is often more likely to be semantically connected to its adjacent sentences than to those that are farther apart. To the best of our knowledge, however, there is a lack of literature on semantic classification methods that take into account both adjacent and distant (non-adjacent) sentences. One example of such an application is prompt pruning for LLMs. When users write prompts, the sentences typically follow a logical flow, but some may be redundant. Splitting a prompt into groups, potentially non-sequential, while still accounting for the semantic coherence of contiguous sentences can facilitate downstream tasks such as prompt pruning, ultimately improving both efficiency and relevance in LLM interactions [6, 25].

In this work, we propose a new framework for text segmentation that accounts for the fact that adjacent sentences are typically more related, while also enabling the grouping of non-contiguous sentences

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Structured Probabilistic Inference & Generative Modeling.

that are semantically similar. To achieve this, we first embed sentences into vector representations using sentence embeddings [21], so that semantically similar sentences are mapped closer together in the embedding space. This allows us to form a graph from the text, where the nodes represent the embedded sentences and edges encode the strength of their semantic relationships. We then apply Belief Propagation (BP, Pearl [19]), an inference algorithm used in graphical models, to generate clusters. To the best of our knowledge, this work presents the first successful application of BP to text segmentation that accounts for semantic meaning of sentences in both continuous and non-continuous settings.

2 Method

Our algorithm, named `BP-Seg`, consists of three main steps: sentence embedding, constructing the graphical model, and running BP. We discuss each of these steps in detail below.

2.1 Sentence embeddings

Given a text, represented as an ordered collection of sentences $\{S_i\}_{i=1}^n$, we can obtain their numerical vector representations using sentence embeddings. This can be efficiently achieved with libraries such as `transformers`, `sentence-transformers`, or `tensorflow_hub`. Once encoded, semantically similar sentences are expected to have higher cosine similarity scores, indicating their closeness in the embedding space. We use R_i to denote the sentence embedding of S_i.

2.2 Constructing the graphical model

The text segmentation process begins with the initialization of a set of cluster representatives, denoted as C_j, which serve as the reference representatives for segment assignments. In practice, these representatives are randomly selected from the set of input sentence embeddings. Given a text with n sentences, we define k clusters and randomly choose k sentence embeddings as the initial representatives, $C = \{C_1, C_2, \ldots, C_k\}$, $C_j \in \{R_1, R_2, \ldots, R_n\}$, with $C_i \neq C_j$. Let $x = \{x_1, x_2, \ldots, x_n\}$ be the segment assignments, where each x_i represents the segment label assigned to sentence S_i. Therefore, each x_i takes a discrete value from the set $\{1, 2, \ldots, k\}$, where k is the total number of segments.

Let $p_i(x_i)$ be the probability that the ith segment is assigned with a label x_i. If we assume the joint distribution factorizes, we can write $p(x_1, \ldots, x_n) = \frac{1}{Z} \prod_f \psi_f(x_f) \prod_g \psi_g(x_g)$, where ψ_f represents unary factors and ψ_g represents pairwise factors, and Z is the normalizing constant. To be precise, $\psi_f(x_f)$ can be written in the form of $\psi_i(x_i)$, encoding how strongly the ith segment prefers the cluster $C_{x_i} \in C$. Similarly, the pairwise factors, $\psi_g(x_g)$ written in the form of $\psi_{i,j}(x_i, x_j)$, encode how compatible the i-th segment is assigned with the label x_i and the j-th segment is assigned with the label x_j, where $x_i, x_j \in \{1, 2, \cdots, k\}$. If $\psi_{i,j}(x_i, x_j)$ is large, it means that assigning the i-th segment with label x_i and the j-th segment with label x_j fits together well. In practice, one has the freedom to choose ψ_f and ψ_g. In this work, we set the node and edge factors as follows:

$$\psi_i(x_i) = \exp(\mathrm{sim}(R_i, C_{x_i})), \tag{1}$$

and

$$\psi_{i,j}(x_i, x_j) = \begin{cases} 1, & x_i = x_j \\ \exp(\lambda\,(\mathrm{sim}(R_i, R_j) - 1)) & \text{otherwise} \end{cases}. \tag{2}$$

Here, $\mathrm{sim}(\cdot, \cdot)$ denotes the cosine similarity between two embeddings. A higher value of $\psi_i(x_i)$ indicates a greater likelihood that sentence S_i belongs to segment x_i. Note also that $\psi_{i,j}(x_i, x_j) \leq 1$, with equality holding if and only if $x_i = x_j$, i.e., when the two sentences are assigned the same label. On the other hand, smaller values of $\psi_{i,j}$ reflect weaker semantic connections between sentences that differ in meaning.

In this work, we adopt specific forms of ψ_i and $\psi_{i,j}$ to encode the semantic relationships between sentences and their assignments. However, in practice, one may choose alternative or domain-specific formulations, provided they are compatible with the desired inference algorithm.

2.3 BP (Sum-Product) for text segmentation

After assigning the node and edges with proper weights, we can start implementing the BP algorithm. The goal of BP is to maximize the marginal probability of segment assignments by iteratively exchanging messages between sentences and updating their segment beliefs. A message from i to j represents node i's belief about the possible values of that node j takes, considering all evidence except what comes from node j itself. At each iteration, every node (sentence embedding) R_i sends a message to its neighboring node R_j, conveying how strongly R_j is associated with a given segment. These messages incorporate both the unary factor, which measures the semantic similarity of a sentence to its assigned segment representative, and the pairwise factor, which enforces consistency between related sentences. Before process begins, the messages $m_{i\to j}(x_j)$ must be initialized. The simplest approach is to set all messages to be uniform, i.e., $m_{i\to j}(x_j) = 1/k$ for all i, j. This assumes no prior preference for any segment, allowing BP to refine the segmentation purely based on updates. The message from R_i to R_j at iteration t is updated as:

$$m_{i\to j}^{(t)}(x_j) = \sum_{x_i} \Big(\psi_i(x_i)\ \psi_{i,j}(x_i, x_j) \prod_{k\in\{1,\dots,n\}\setminus j} m_{k\to i}^{(t-1)}(x_i)\Big), \tag{3}$$

where ψ_i is the unary potential, and $\psi_{i,j}$ is the pairwise potential. Each node updates its belief about its segment assignment by accumulating incoming messages from all neighboring sentences:

$$b_i(x_i) \ \propto\ \psi_i(x_i) \prod_{j\in\{1,\dots,n\}\setminus i} m_{j\to i}(x_i). \tag{4}$$

This iterative process continues until convergence, where the segment labels stabilize. The final segmentation is determined by selecting the segment with the highest belief for each node,

$$x_i^* = \underset{x_i\in\{1,\cdots,k\}}{\arg\max}\ b_i(x_i). \tag{5}$$

We summarize our proposed algorithm in Algorithm 1. More analysis of BP can be read in Murphy et al. [17], Yedidia et al. [26].

Algorithm 1 `BP-Seg`

Input: Sentence embeddings $\{R_1, \dots, R_n\}$
Output: Segment assignment $\{x_1^*, \dots, x_n^*\}$
Initialization: Initialize k segment representatives $\{C_1, \dots, C_k\}$. Initialize node and edge factors following (1) and (2), and initialize all messages $m_{i\to j}^{(0)}(x_j) = 1/k$.
for $t = 1$ to T **do**
  for each embedding R_i **do**
    for each $R_j \in \{1, \dots, n\} \setminus i$ **do**
      Update messages using (3).
    end for
  end for
end for
for each embedding R_i **do**
  Update belief using (4).
end for
Final Segmentation:
for each embedding R_i **do**
  Assign segment with highest belief using (5).
end for
return Segment assignments $\{x_1^*, \dots, x_n^*\}$

3 Related work

A prior work has explored a variation of using graph-based models for text segmentation. `GraphSeg`, proposed by Glavaš et al. [8], for example, also employs unsupervised learning for text segmentation within a graph-based framework. However, their primary objective is to produce *contiguous*

	3–5		6–8		9–11		3–11		3–15		12–15	
	ARI	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI	NMI
`BP-Seg`	0.58	0.83	**0.76**	**0.89**	**0.73**	**0.87**	**0.73**	**0.87**	**0.65**	**0.84**	**0.62**	**0.83**
`GraphSeg`	**0.65**	**0.87**	0.58	0.83	0.52	0.81	0.55	0.83	0.46	0.79	0.40	0.77
k-means	0.53	0.84	0.52	0.79	0.52	0.76	0.50	0.79	0.45	0.74	0.45	0.70

Table 1: Average performance on the Choi dataset measured using ARI and NMI across different subsets. Higher values indicate better performance. Note that a random segmentation method achieves a 0 in ARI.

segmentations, whereas our method allows for a *non-contiguous* segmentation that accounts for both neighboring and distant sentences. Moreover, their algorithm requires additional information, such as each word's information content based on its relative frequency, whereas `BP-seg` relies *solely* on embeddings and no external data. Moreover, after encoding sentences into embeddings and computing cosine similarities, their approach discovers segmentations by finding maximal cliques — fundamentally different from our *probabilistic* strategy, in which we seek an assignment that maximizes the marginal distribution. One could in principle apply k-means to group sentence embeddings; however, such a method is entirely context-agnostic and considers only pairwise embedding similarities.

4 Experiments

4.1 The Illustrative Example

To demonstrate the effectiveness of our proposed segmentation method, we compare its performance against `GraphSeg` (implemented using the code available here[1]), k-means, and a large language model (LLM). The input text used for segmentation, generated by GPT-4o with additional human-written content, is as follows:

```
The sun was shining brightly. It was a beautiful morning. I decided to
go for a walk. Suddenly, dark clouds appeared. I'll play tennis tomorrow.
What are you doing? Thunder rumbled in the distance. The rain poured down
heavily. People ran for shelter. US Open is a tennis tournament. I am
here working on my project. The sun came out again. Who is going to win
the US Open?
```

Each sentence in the document is treated as an individual unit for segmentation. We evaluate the segmentation outcomes generated by our proposed method, `BP-Seg`, in comparison with three baselines: `GraphSeg`, k-means clustering, and a LLM (GPT-4o). To qualitatively assess the results, we focus on two key criteria: (1) whether the method identifies more than one meaningful segment (i.e., produces more than one cluster), and (2) whether the three semantically related tennis sentences, highlighted in red, are grouped together in the same cluster. These criteria help determine both the model's ability to detect structure and its sensitivity to semantic coherence. The full experimental setup, including parameter choices and additional visualizations across methods and configurations, can be found in Appendix A.1.

For `BP-Seg` and k-means, we set the number of segments to $k = 2, 3, \cdots, 7$. For the LLM, we explicitly prompt it to generate $k = 2, 3, \cdots, 7$ segments. The prompt can be read in A.1.4. As observed, all methods generate more than one clusters. Furthermore, k-means successfully group sentences based on thematic coherence across all values of k. Both `BP-Seg` and the LLM successfully group the tennis-related sentences in the same cluster in 5 out of the 6 tested values of k. For `GraphSeg`, we set the minimum number of sentences per segment to 1 to allow maximum flexibility. However, despite trying various thresholds τ, `GraphSeg` fails to cluster semantically related sentences effectively, as it is designed for contiguous segmentation. We summarize the results in Table 2.

4.2 Choi dataset

In this example, we implement our approach, `BP-Seg`, along with `GraphSeg` and k-means on the Choi dataset [5]. The performance of the LLM is not included in this case, as its output does not

[1] `https://github.com/Dobatymo/graphseg-python`

	2 Clusters	3 Clusters	4 Clusters	5 Clusters	6 Clusters	7 Clusters
`BP-Seg`	✓	✓	✓	✗	✓	✓
k-means	✓	✓	✓	✓	✓	✓
LLM	✓	✓	✓	✓	✗	✓
`Graph-Seg`	✗	✗	✗	✗	✗	✗

Table 2: Performance on an illustrative example. Both `BP-Seg` and k-means are able to group tennis-related sentences into the same clusters, with varying numbers of clusters.

include every sentence from the original text. Traditionally, for contiguous text segmentation, two evaluation metrics are commonly reported: P_k [3] and WindowDiff (WD) [20]. The P_k metric checks whether the boundary status (i.e., whether two sentences within a fixed-size window belong to the same segment) matches between the ground truth and the prediction. WD, on the other hand, measures whether the number of boundaries within the window is consistent with the ground truth. However, both metrics assume contiguity and are not suitable for evaluating non-contiguous text segmentation.

Therefore, we report Adjusted Rand Index (ARI) [11, 23] and Normalized Mutual Information (NMI) [13], which are appropriate for clustering-based evaluations. This adjustment accounts for the fact that the outputs of `BP-Seg` and k-means may result in non-contiguous segmentations, even though the ground truth segmentation is contiguous. Additionally, for efficiency, we use a variant of `BP-Seg` that is also based on message passing but offers faster computation. Please refer to Algorithm 2 for more details.

The ARI and Normalized NMI are two widely used metrics for evaluating clustering and segmentation quality. ARI ranges from -1 to 1, where a score of 1 indicates a perfect agreement between the predicted segmentation and the ground truth, 0 reflects a performance equivalent to random labeling, and negative values suggest an agreement worse than random chance. This makes ARI particularly informative in distinguishing between meaningful segmentations and those produced by chance. NMI, on the other hand, measures the mutual dependence between the predicted and true labels. It ranges from 0 to 1, with 1 indicating complete alignment between the predicted and true segmentations, and 0 representing statistical independence. Unlike ARI, NMI is insensitive to permutations of cluster labels, making it a complementary metric for evaluating clustering performance.

Table 1 reports the mean segmentation performance of the three methods on the Choi dataset. As illustrated in the table, `BP-Seg` consistently outperforms the baseline methods across nearly all configurations, except in cases where each segment contains very few sentences (e.g., 3–5). This highlights the strength of our belief propagation-based formulation in capturing both local and global textual coherence. The standard deviation of the performance can be read in Table 3.

5 Conclusion

We presented `BP-Seg`, an efficient unsupervised approach for text segmentation using belief propagation. Our method effectively balances local contextual coherence with global semantic similarity, enabling more meaningful and flexible segmentation of text. Although designed for non-contiguous segmentation, experimental results show that `BP-Seg` outperforms several competitive methods on the standard contiguous segmentation task, achieving strong performance on metrics such as ARI and NMI. Looking forward, we aim to explore the utility of `BP-Seg` in real-world downstream applications. These include prompt pruning for LLMs, where segmenting prompts into semantically coherent chunks can lead to more efficient inference; information retrieval, where flexible segmentation can support better indexing and matching; and question answering, where isolating relevant spans is crucial for interpretability and performance. We believe the probabilistic and modular nature of `BP-Seg` offers a promising foundation for broader integration into these complex language understanding pipelines.

Limitations

In this study, all examples are in English. The example in Section 4.1 was generated by GPT-4o with additional human-written content, and the Choi dataset in Section 4.2 is also synthetic. As a result, these examples may not accurately reflect real-world scenarios, and our evaluations are limited to these two cases. Nevertheless, we believe the insights from our findings will inspire further research in text segmentation and benefit a wide range of related applications.

References

[1] Pinkesh Badjatiya, Litton J. Kurisinkel, Manish Gupta, and Vasudeva Varma. Attention-based neural text segmentation. *ArXiv*, abs/1808.09935, 2018. URL https://api.semanticscholar.org/CorpusID:4090560.

[2] Berat Kurar Barakat, Ahmad Droby, Reem Alaasam, Boraq Madi, Irina Rabaev, Raed Shammes, and Jihad El-Sana. Unsupervised deep learning for text line segmentation. *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 2304–2311, 2020. URL https://api.semanticscholar.org/CorpusID:221663960.

[3] Doug Beeferman, Adam Berger, and John Lafferty. Statistical models for text segmentation. *Machine learning*, 34:177–210, 1999.

[4] Sangwoo Cho, Kaiqiang Song, Xiaoyang Wang, Fei Liu, and Dong Yu. Toward unifying text segmentation and long document summarization. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 106–118, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.8. URL https://aclanthology.org/2022.emnlp-main.8/.

[5] Freddy YY Choi. Advances in domain independent linear text segmentation. *arXiv preprint cs/0003083*, 2000.

[6] Jun Gao, Ziqiang Cao, and Wenjie Li. Selfcp: Compressing over-limit prompt via the frozen large language model itself. *Information Processing & Management*, 61:103873, 11 2024. doi: 10.1016/j.ipm.2024.103873.

[7] Goran Glavas and Swapna Somasundaran. Two-level transformer and auxiliary coherence modeling for improved text segmentation. *ArXiv*, abs/2001.00891, 2020. URL https://api.semanticscholar.org/CorpusID:209832486.

[8] Goran Glavaš, Federico Nanni, and Simone Paolo Ponzetto. Unsupervised text segmentation using semantic relatedness graphs. In Claire Gardent, Raffaella Bernardi, and Ivan Titov, editors, *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, pages 125–130, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/S16-2016. URL https://aclanthology.org/S16-2016/.

[9] Marti A. Hearst. Texttiling: segmenting text into multi-paragraph subtopic passages. 23(1): 33–64, March 1997. ISSN 0891-2017.

[10] Xiangji Huang, Fuchun Peng, Dale Schuurmans, Nick Cercone, and Stephen Robertson. Applying machine learning to text segmentation for information retrieval. *Inf. Retr.*, 6:333–362, 09 2003. doi: 10.1023/A:1026028229881.

[11] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2:193–218, 1985.

[12] Omri Koshorek, Adir Cohen, Noam Mor, Michael Rotman, and Jonathan Berant. Text segmentation as a supervised learning task. *ArXiv*, abs/1803.09337, 2018. URL https://api.semanticscholar.org/CorpusID:4411469.

[13] Tarald O Kvålseth. On normalized mutual information: measure derivations and properties. *Entropy*, 19(11):631, 2017.

[14] Mike Lewis and Tom Brown. Retrieval-augmented generation for large language models. *arXiv preprint arXiv:2312.10997*, 2023. URL https://arxiv.org/pdf/2312.10997.

[15] Stuart P Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982. doi: 10.1109/TIT.1982.1056489.

[16] Lesly Miculicich and Benjamin Han. Document summarization with text segmentation. *ArXiv*, abs/2301.08817, 2023. URL https://api.semanticscholar.org/CorpusID:256105151.

[17] Kevin Murphy, Yair Weiss, and Michael I Jordan. Loopy belief propagation for approximate inference: An empirical study. *arXiv preprint arXiv:1301.6725*, 2013.

[18] Irina Pak and Phoey Lee Teh. Text segmentation techniques: A critical review. 2018. URL https://api.semanticscholar.org/CorpusID:196033032.

[19] Judea Pearl. Reverend bayes on inference engines: A distributed hierarchical approach. *Probabilistic and Causal Inference*, 1982. URL https://api.semanticscholar.org/CorpusID:14936636.

[20] Lev Pevzner and Marti A. Hearst. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, 28(1):19–36, 2002. doi: 10.1162/089120102317341756. URL https://aclanthology.org/J02-1002/.

[21] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. pages 3973–3983, 01 2019. doi: 10.18653/v1/D19-1410.

[22] Shiwon Song. The informational value of segment data disaggregated by underlying industry: Evidence from the textual features of business descriptions. *The Accounting Review*, 96(6): 361–396, 2021.

[23] Silke Wagner and Dorothea Wagner. Comparing clusterings: an overview. 2007.

[24] Yizhong Wang, Sujian Li, and Jingfeng Yang. Toward fast and accurate neural discourse segmentation. In *Conference on Empirical Methods in Natural Language Processing*, 2018. URL https://api.semanticscholar.org/CorpusID:52115097.

[25] Zhentao Xu, Fengyi Li, Albert Chen, and Xiaofeng Wang. Procut: Llm prompt compression via attribution estimation, 2025. URL https://arxiv.org/abs/2508.02053.

[26] Jonathan S Yedidia, William T Freeman, Yair Weiss, et al. Understanding belief propagation and its generalizations. *Exploring artificial intelligence in the new millennium*, 8(236–239): 0018–9448, 2003.

[27] Hai Yu, Chong Deng, Qinglin Zhang, Jiaqing Liu, Qian Chen, and Wen Wang. Improving long document topic segmentation models with enhanced coherence modeling. *ArXiv*, abs/2310.11772, 2023. URL https://api.semanticscholar.org/CorpusID:264289093.

A Appendix

A.1 The Illustrave Example

A.1.1 Results using `BP-Seg`

Set $\lambda = 0.12$ **in** (2)
$k = 2$ **segments**: [Segment 1]: The sun was shining brightly. It was a beautiful morning. I decided to go for a walk. Suddenly, dark clouds appeared. I'll play tennis tomorrow. What are you doing? The rain poured down heavily. People ran for shelter. US Open is a tennis tournament. I am here working on my project. The sun came out again. Who is going to win the US Open? [Segment 2]: Thunder rumbled in the distance.

$k = 3$ **segments**: [Segment 1]: The sun was shining brightly. It was a beautiful morning. I decided to go for a walk. Suddenly, dark clouds appeared. Thunder rumbled in the distance. The rain poured down heavily. The sun came out again. [Segment 2]: I'll play tennis tomorrow. What are you doing? US Open is a tennis tournament. I am here working on my project. Who is going to win the US Open? [Segment 3]: People ran for shelter.

$k = 4$ **segments**: [Segment 1]: The sun was shining brightly. It was a beautiful morning. I decided to go for a walk. What are you doing? People ran for shelter. I am here working on my project. The sun came out again. [Segment 2]: Suddenly, dark clouds appeared. The rain poured down heavily. [Segment 3]: Thunder rumbled in the distance. [Segment 4]: I'll play tennis tomorrow. US Open is a tennis tournament. Who is going to win the US Open?

$k = 5$ **segments**: [Segment 1]: The sun was shining brightly. It was a beautiful morning. Suddenly, dark clouds appeared. Thunder rumbled in the distance. People ran for shelter. US Open is a tennis tournament. The sun came out again. [Segment 2]: I decided to go for a walk. [Segment 3]: I'll play tennis tomorrow. What are you doing? Who is going to win the US Open? [Segment 4]: The rain poured down heavily. [Segment 5]: I am here working on my project.

$k = 6$ **segments**: [Segment 1]: The sun was shining brightly. It was a beautiful morning. The rain poured down heavily. The sun came out again. [Segment 2]: I decided to go for a walk. [Segment 3]: I'll play tennis tomorrow. US Open is a tennis tournament. Who is going to win the US Open? [Segment 4]: Suddenly, dark clouds appeared. Thunder rumbled in the distance. [Segment 5]: People ran for shelter. [Segment 6]: What are you doing? I am here working on my project.

$k = 7$ **segments**: [Segment 1]: The sun was shining brightly. People ran for shelter. [Segment 2]: It was a beautiful morning. I decided to go for a walk. [Segment 3]: Suddenly, dark clouds appeared. Thunder rumbled in the distance. [Segment 4]: What are you doing? I am here working on my project. [Segment 5]: The rain poured down heavily. [Segment 6]: I'll play tennis tomorrow. US Open is a tennis tournament. Who is going to win the US Open? [Segment 7]: The sun came out again.

A.1.2 Results using `GraphSeg`

treshold $\tau = 0.1, 0.3, 0.5, 0.7, 0.9$ **and minimal segment size** $n = 1$: [Segment 1]: The sun was shining brightly. It was a beautiful morning. [Segment 2]: I decided to go for a walk. Suddenly, dark clouds appeared. What are you doing? [Segment 3]: Thunder rumbled in the distance. The rain poured

down heavily. [Segment 4]: People ran for shelter. US Open is a tennis tournament. [Segment 5]: I am here working on my project. The sun came out again. Who is going to win the US Open?

A.1.3 Results using k-means

$k = 2$ **segments**: [Segment 1]: I'll play tennis tomorrow. What are you doing? US Open is a tennis tournament. I am here working on my project. Who is going to win the US Open? [Segment 2]: The sun was shining brightly. It was a beautiful morning. I decided to go for a walk. Suddenly, dark clouds appeared. Thunder rumbled in the distance. The rain poured down heavily. People ran for shelter. The sun came out again.

$k = 3$ **segments**: [Segment 1]: The sun was shining brightly. It was a beautiful morning. Suddenly, dark clouds appeared. Thunder rumbled in the distance. The rain poured down heavily. People ran for shelter. The sun came out again. [Segment 2]: I'll play tennis tomorrow. US Open is a tennis tournament. Who is going to win the US Open? [Segment 3]: I decided to go for a walk. What are you doing? I am here working on my project.

$k = 4$ **segments**: [Segment 1]: The sun was shining brightly. It was a beautiful morning. Suddenly, dark clouds appeared. Thunder rumbled in the distance. The rain poured down heavily. The sun came out again. [Segment 2]: I'll play tennis tomorrow. US Open is a tennis tournament. Who is going to win the US Open? [Segment 3]: I decided to go for a walk. What are you doing? I am here working on my project. [Segment 4]: People ran for shelter.

$k = 5$ **segments**: [Segment 1]: I decided to go for a walk. People ran for shelter. [Segment 2]: The sun was shining brightly. Suddenly, dark clouds appeared. Thunder rumbled in the distance. The sun came out again. [Segment 3]: What are you doing? I am here working on my project. [Segment 4]: I'll play tennis tomorrow. US Open is a tennis tournament. Who is going to win the US Open? [Segment 5]: It was a beautiful morning. The rain poured down heavily.

$k = 6$ **segments**: [Segment 1]: Suddenly, dark clouds appeared. Thunder rumbled in the distance. The rain poured down heavily. [Segment 2]: What are you doing? I am here working on my project. [Segment 3]: I'll play tennis tomorrow. US Open is a tennis tournament. Who is going to win the US Open? [Segment 4]: The sun was shining brightly. The sun came out again. [Segment 5]: People ran for shelter. [Segment 6]: It was a beautiful morning. I decided to go for a walk.

$k = 7$ **segments**: [Segment 1]: I am here working on my project. [Segment 2]: The sun was shining brightly. It was a beautiful morning. The sun came out again. [Segment 3]: I'll play tennis tomorrow. US Open is a tennis tournament. Who is going to win the US Open? [Segment 4]: I decided to go for a walk. What are you doing? [Segment 5]: People ran for shelter. [Segment 6]: Suddenly, dark clouds appeared. Thunder rumbled in the distance. [Segment 7]: The rain poured down heavily.

A.1.4 Results using LLM (GPT-4o)

Prompt: Please generate {number_seg} segmentation for the following text according to their sematic meaning. Adjacent sentences do not have to be in the same segment. The sun was shining brightly. It was a beautiful morning. I decided to go for a walk. Suddenly, dark clouds appeared.

I'll play tennis tomorrow. What are you doing? Thunder rumbled in the distance. The rain poured down heavily. People ran for shelter. US Open is a tennis tournament. I am here working on my project. The sun came out again. Who is going to win the US Open? Return the output in the following format: [Segment 1]: The sun was shining brightly. It was a beautiful morning. [Segment 2]: I decided to go for a walk. Suddenly, dark clouds appeared. ...

$k = 2$ **segments**: [Segment 1]: The sun was shining brightly. It was a beautiful morning. Suddenly, dark clouds appeared. Thunder rumbled in the distance. The rain poured down heavily. People ran for shelter. The sun came out again. [Segment 2]: I decided to go for a walk. I'll play tennis tomorrow. What are you doing? US Open is a tennis tournament. I am here working on my project. Who is going to win the US Open?

$k = 3$ **segments**: [Segment 1]: The sun was shining brightly. It was a beautiful morning. The sun came out again. [Segment 2]: I decided to go for a walk. Suddenly, dark clouds appeared. Thunder rumbled in the distance. The rain poured down heavily. People ran for shelter. [Segment 3]: I'll play tennis tomorrow. US Open is a tennis tournament. Who is going to win the US Open?

$k = 4$ **segments**: [Segment 1]: The sun was shining brightly. It was a beautiful morning. The sun came out again. [Segment 2]: I decided to go for a walk. Suddenly, dark clouds appeared. Thunder rumbled in the distance. The rain poured down heavily. People ran for shelter. [Segment 3]: I'll play tennis tomorrow. US Open is a tennis tournament. Who is going to win the US Open? [Segment 4]: What are you doing? I am here working on my project.

$k = 5$ **segments**: [Segment 1]: The sun was shining brightly. It was a beautiful morning. The sun came out again. [Segment 2]: I decided to go for a walk. Suddenly, dark clouds appeared. Thunder rumbled in the distance. The rain poured down heavily. People ran for shelter. [Segment 3]: I'll play tennis tomorrow. US Open is a tennis tournament. Who is going to win the US Open? [Segment 4]: What are you doing? [Segment 5]: I am here working on my project.

$k = 6$ **segments**: [Segment 1]: The sun was shining brightly. It was a beautiful morning. [Segment 2]: I decided to go for a walk. Suddenly, dark clouds appeared. Thunder rumbled in the distance. The rain poured down heavily. People ran for shelter. The sun came out again. [Segment 3]: I'll play tennis tomorrow. [Segment 4]: What are you doing? [Segment 5]: US Open is a tennis tournament. Who is going to win the US Open? [Segment 6]: I am here working on my project.

$k = 7$ **segments**: [Segment 1]: The sun was shining brightly. It was a beautiful morning.[Segment 2]: I decided to go for a walk. Suddenly, dark clouds appeared. [Segment 3]: Thunder rumbled in the distance. The rain poured down heavily. People ran for shelter. [Segment 4]: The sun came out again. [Segment 5]: I'll play tennis tomorrow. US Open is a tennis tournament. Who is going to win the US Open? [Segment 6]: What are you doing? [Segment 7]: I am here working on my project.

A.2 Choi dataset

Here $m_i(x_i)$ can be viewed as a running preference or partial belief for node i in label x_i. We also set $\psi_i(x_i) = \mathrm{sim}(R_i, C_{x_i})$ and $\psi_{i,j} = \lambda \mathrm{sim}(R_i, R_j) \exp(-|i-j|^2/\sigma)$. For comparison, we set the number of initial clusters in BP-Seg to be equal to the number of sentences, although many clusters

may become empty after the algorithm terminates. We fix the number of iterations to $T = 5$, and choose $\sigma = 10$ and $\lambda = 300$ to account for the length of the text. For `GraphSeg`, we set the threshold parameter to $\tau = 0.2$ and the minimum segment size to $n = 1$ to allow maximum flexibility. For the k-means baseline, we cap the number of clusters k at 20 and rely on the default number of iterations in `sklearn.cluster.KMeans`.

Algorithm 2 Fast `BP-Seg`

Input: Sentence embeddings $\{R_1, \ldots, R_n\}$
Output: Segment assignment $\{x_1^*, \ldots, x_n^*\}$
Initialization: Initialize k segment representatives $\{C_1, \ldots, C_k\}$. Initialize node and edge factors following (1) and (2), and initialize all messages $m_i^{(0)}(x_i) = 1/k$.
for $t = 1$ to T **do**
  for each embedding R_i **do**
    for $x_i = \{1, 2, \cdots, k\}$ **do**
      $m_i^{(t)}(x_i) = \psi_i(x_i) + \sum_{j=1}^{n} \psi_{i,j} m_j^{(t-1)}(x_i)$
    end for
  end for
end for
for each embedding R_i **do**
  Update belief using $b_i(x_i) = \psi_i(x_i) + m_i^{(T)}(x_i)$
end for
Final Segmentation:
for each embedding R_i **do**
  Assign segment using $x_i^* = \arg\max_{x_i \in \{1,\cdots,k\}} b_i(x_i)$.
end for
return Segment assignments $\{x_1^*, \ldots, x_n^*\}$

	3–5		**6–8**		**9–11**		**3–11**		**3–15**		**12–15**	
	ARI	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI	NMI
`BP-Seg`	0.10	0.04	0.08	0.04	0.08	0.04	0.09	0.04	0.08	0.03	0.08	0.03
`GraphSeg`	0.11	0.04	0.07	0.03	0.06	0.02	0.08	0.03	0.06	0.02	0.07	0.02
k-means	0.09	0.03	0.08	0.04	0.08	0.05	0.08	0.04	0.07	0.04	0.07	0.05

Table 3: Standard deviation on the Choi dataset measured using ARI and NMI across different subsets.