BART-Dijkstra-SA:

A Scalable Workflow for Semantic Summarization and Topic Structuring of Large-Scale Text Data

Author: Chen-Hong
Affiliation: National Chung Cheng University
Date: April 26, 2025

Abstract

As the global volume of information grows exponentially, the management, comprehension, and generation of large-scale text data have become central challenges in the fields of Natural Language Processing (NLP) and big data analytics. To address the limitations of existing generative models in thematic focus and semantic consistency, this study proposes the BART-Dijkstra-SA (BDS) workflow as a systematic solution for large-scale text topic modeling and summarization.
The BDS workflow integrates simulated annealing for optimizing the number of topics, employs Latent Dirichlet Allocation (LDA) and TF-IDF vectorization to construct semantic graphs, and uses the Leiden algorithm to ensure community connectivity and modularity stability.
Subsequently, Dijkstra's shortest path algorithm is applied over sparse semantic graphs to extract semantically coherent keyword sequences, serving as structural prompts that guide the BART model to generate thematically focused and semantically consistent summaries and titles.
The proposed method demonstrates excellent scalability and stable generative performance as the sample size increases, making it suitable for large-scale topic modeling, summarization, and knowledge management, providing a next-generation systematic solution for natural language understanding and generation technologies.

---

1. Introduction

The exponential growth of global information has rendered the management, understanding, and generation of large-scale text data a core challenge in natural language processing (NLP) and big data analytics (Reuters Institute for the Study of, 2023; Silverman, 2016).
In application contexts such as misinformation analysis, news monitoring, and topic modeling, text data often exhibits complex and intertwined semantic structures, placing stricter demands on generative models in thematic focus, semantic consistency, and structural comprehension.

Mainstream generative pre-trained models such as BART (Lewis et al., 2019) demonstrate strong language understanding and generation capabilities.
However, due to the $O(n^2)$ computational complexity of the self-attention

mechanism, they are prone to semantic drift and thematic defocusing when handling million-scale samples or multi-themed long texts.
Furthermore, semantic graph construction methods such as TF-IDF vectorization and cosine similarity calculations also exhibit $O(k^2)$ complexity when expanding keyword sets, leading to performance bottlenecks.
Traditional community detection methods like the Louvain algorithm (Blondel et al., 2008), although capable of quickly approximating modularity maximization, often face issues such as community fragmentation and clustering distortion under large-scale conditions (Traag et al., 2019), and their O(n log n) complexity is insufficient to meet ultra-large-scale semantic network processing needs.

In big data environments, the design of data structures, control of algorithmic time complexity, and workflow automation have become critical factors in determining whether a text generation and comprehension system is scalable.
Traditional techniques relying on manual parameter tuning, brute-force searches, or unguided generation cannot maintain computational efficiency and output quality as the sample size n grows rapidly, leading to resource bottlenecks and performance degradation.

In response to these challenges, this study proposes the **BART-Dijkstra-SA workflow (BDS)** as a systematic solution for processing large-scale text data.
The workflow integrates three core techniques: structural modeling, heuristic optimization, and structure-guided generation, specifically:

1. **Topic Number Optimization (Simulated Annealing, SA):**
   Utilizing simulated annealing (Kirkpatrick et al., 1983) with perplexity as the objective function to automatically explore the topic number space. Compared to traditional grid search ($O(k \cdot n)$), simulated annealing reduces complexity to approximately $O(\log k \cdot n)$, significantly alleviating computational burdens in large datasets.
2. **Topic Modeling and Semantic Graph Construction (LDA + TF-IDF + Cosine Similarity):**
   Conducting LDA topic modeling based on the optimized topic number, and constructing a keyword semantic graph using TF-IDF vectorization and cosine similarity as a foundation for subsequent structural analysis.
3. **Community Detection (Leiden Algorithm):**
   Employing the Leiden algorithm for community clustering ensures the connectivity and modularity stability of semantic graphs, maintaining time complexity at $O(m)$ (where m is the number of edges).

4. **Shortest Semantic Path Search (Heap-Optimized Dijkstra Algorithm):**
   For each topic community, employing the Dijkstra algorithm over sparse semantic graphs with heap optimization, effectively reducing the complexity to O(E + V log V).
5. **Structure-Guided Generation (BART Model):**
   Using the extracted shortest semantic paths as structured prompts to guide the BART model in generating thematically focused, semantically coherent summaries and titles.

Overall, the BART-Dijkstra-SA workflow balances deep structural modeling, controllable computational complexity, and stable generation quality.
Its scalability becomes increasingly evident as sample size n expands, demonstrating strong potential for large-scale text analysis, summarization, and knowledge management applications.

2. Related Work

2.1 Text Generation and Semantic Modeling Techniques

Text generation technologies have continuously evolved in the field of natural language processing (NLP), particularly with the development of generative pre-trained models, which have greatly enhanced machine capabilities in language understanding and generation.
The BART (Bidirectional and Auto-Regressive Transformers) model (Lewis et al., 2019) combines encoder and decoder architectures and is trained via denoising autoencoder tasks, enabling the model to effectively repair corrupted text and perform high-quality generation tasks.
Its training method simulates corruption operations such as masking, shuffling, and segment deletion, thus strengthening the model's ability to grasp semantic structures, making BART a critical tool in tasks like summarization, translation, and inference.

On the other hand, topic modeling serves as a key technique for understanding the thematic structure of text.
The Latent Dirichlet Allocation (LDA) model (Blei et al., 2003) is widely applied in large-scale text analysis.

LDA assumes that documents are mixtures of several latent topics, each described by a distribution over words, allowing hidden thematic structures in the text to be automatically revealed.
However, LDA is sensitive to the number of topics set, and it can easily fall into local optima when faced with complex topic spaces (Pathik & Shukla, 2020).

In terms of semantic graph construction, TF-IDF (Term Frequency–Inverse Document Frequency) and cosine similarity are commonly used methods to measure the similarity between words, effectively quantifying semantic proximity between text elements (Zhang et al., 2011).
Such techniques support the construction of text or keyword graphs, providing a foundational basis for subsequent semantic path search and structural analysis.

Nevertheless, as the number of samples and topics increases rapidly, existing methods still face challenges in processing efficiency and thematic focus.
This study addresses these limitations by proposing a structure-guided text generation workflow to improve thematic consistency and summarization quality.

2.2 Semantic Structure Construction and Community

Detection Techniques

Community detection within semantic graphs is essential for understanding inter-topic relationships and the structural organization of text data.
Traditional methods such as the Louvain algorithm (Blondel et al., 2008) optimize for modularity maximization and can quickly perform clustering with approximately O(n log n) time complexity.
However, subsequent studies have shown that the Louvain method can exhibit issues such as insufficient community connectivity and elevated modularity errors under certain conditions (Traag et al., 2019), resulting in distorted topic structures.

To overcome these shortcomings, the Leiden algorithm (Traag et al., 2019) improves the clustering process by guaranteeing full connectivity within each community and enhancing modular stability while maintaining a time complexity of O(m) (where m represents the number of edges).

The Leiden algorithm thus proves highly suitable for handling the expansion of semantic graphs in big data environments.

Furthermore, to extract the most semantically coherent associative paths between texts or topics, graph theory's Dijkstra algorithm (Dijkstra, 1959) is widely employed. In traditional array-based implementations, Dijkstra's algorithm exhibits $O(V^2)$ time complexity.
To enhance efficiency, this study adopts a sparse semantic graph structure and employs a heap-optimized version of Dijkstra's algorithm, reducing the complexity to $O(E + V \log V)$.
This strategy enables efficient extraction of the shortest semantic paths in scenarios where the number of topic keywords remains moderate.

In designing the methodology, this study particularly emphasizes the integration of clustering accuracy, path search efficiency, and semantic consistency as critical considerations.

2.3 Large-Scale Text Processing and Time Complexity

Control

Under the background of big data, text data volumes are increasing exponentially. Traditional brute-force computations or non-optimized workflows inevitably face severe resource bottlenecks and efficiency issues (Kannan et al., 2016).
Therefore, controlling algorithmic time complexity and automating workflows have become key factors determining the scalability of text analysis systems.

In the context of topic modeling, conventional hyperparameter optimization approaches such as random initialization or grid search typically exhibit time complexity of $O(k \cdot n)$ (where k is the number of topics and n is the number of samples), rendering them impractical for large-scale datasets.
To address this, heuristic optimization methods like simulated annealing (Simulated Annealing, SA) (Kirkpatrick et al., 1983) have proven effective in automating topic number selection with near $O(\log k \cdot n)$ efficiency, avoiding local optima and reducing computational loads (Pathik & Shukla, 2020).

Moreover, after constructing the semantic graph and performing community detection, effectively selecting representative prompt keywords for guiding generation becomes crucial for enhancing system efficiency and semantic quality. Through shortest-path-based prompt selection strategies, it is possible to control computational resource consumption while enhancing thematic focus and structural consistency, thus achieving precise and efficient text generation goals.

In summary, the design of this study prioritizes high scalability, low time complexity, and semantic quality stability.
By integrating topic optimization, community clustering, shortest path search, and structure-guided generation techniques, this research proposes the BART-Dijkstra-SA (BDS) workflow as a systematic solution to the challenges of large-scale text understanding and generation.

3. Methodology

3.1 Workflow Overview

To address the challenges of large-scale text topic modeling, summarization, and structural understanding, this study proposes the BART-Dijkstra-SA (BDS) workflow. The workflow integrates three core components: structural modeling, heuristic optimization, and structure-guided generation, aiming to balance semantic focus, computational efficiency, and scalability.
The overall workflow is as follows:

1. **Topic Number Optimization (Simulated Annealing, SA):**
   Automatically optimize the number of topics using simulated annealing, thereby reducing the computational burden compared to manual tuning and brute-force search.
2. **Topic Modeling and Semantic Graph Construction (LDA + TF-IDF + Cosine Similarity):**
   Perform topic modeling using Latent Dirichlet Allocation (LDA) and construct semantic graphs through TF-IDF vectorization and cosine similarity calculations.

3. **Community Detection (Leiden Algorithm):**
   Apply the Leiden algorithm to detect communities, ensuring connectivity and modularity stability among topic clusters.
4. **Shortest Semantic Path Extraction (Heap-Optimized Dijkstra Algorithm):**
   Utilize the heap-optimized Dijkstra algorithm to extract semantically coherent keyword sequences from each topic cluster.
5. **Structure-Guided Generation (BART Model):**
   Use the extracted keyword sequences as structured prompts to guide the BART model in generating thematically focused and semantically coherent summaries and titles.

This workflow is designed to ensure system scalability in big data environments while maintaining semantic quality for natural language generation tasks.

3.2 Topic Number Optimization (Simulated Annealing)

The selection of the number of topics (k) significantly affects the performance of topic modeling.
Traditional grid search methods require traversing multiple candidate parameters, resulting in a time complexity of O(k·n), which becomes computationally prohibitive for large datasets (Hasan et al., 2020).

To address this, this study adopts **Simulated Annealing (SA)** (Kirkpatrick et al., 1983) as the optimization mechanism.
SA simulates the physical annealing process through a controlled stochastic search, effectively reducing the number of search iterations and avoiding local optima.
In this study, the perplexity of the LDA model serves as the energy function.
Through temperature decrement and neighborhood search strategies, the process converges to a near-globally optimal number of topics.
This method reduces the complexity to approximately O(log k·n), significantly enhancing the feasibility of topic modeling in large-scale text datasets.

3.3 Topic Modeling and Semantic Graph Construction

(LDA + TF-IDF + Cosine Similarity)

After optimizing the number of topics, the LDA model (Blei et al., 2003) is employed for topic modeling.
LDA assumes that each document is generated from a mixture of topics, with each topic represented by a probability distribution over words, effectively capturing the latent semantic structures of the text.

To further construct the semantic graph among topic keywords, this study uses TF-IDF vectorization to represent words as vectors and calculates the cosine similarity between them.
Keywords are treated as nodes, and similarity scores as edge weights, forming a semantic graph.
This graph structure serves as the foundation for subsequent community detection and shortest semantic path extraction.

3.4 Community Detection (Leiden Algorithm)

To understand the structural relationships between topics, this study applies community detection to the constructed semantic graph.
While the traditional Louvain algorithm (Blondel et al., 2008) can quickly approximate modularity maximization, it suffers from problems such as community fragmentation and instability (Traag et al., 2019).

To overcome these issues, the **Leiden algorithm** (Traag et al., 2019) is adopted.
The Leiden algorithm refines partitions through detailed optimizations and connectivity checks after each step, ensuring full connectivity within each community and improving modularity stability.
The Leiden algorithm operates with a time complexity of O(m), making it suitable for managing large-scale semantic network structures and supporting downstream shortest path keyword extraction tasks.

3.5 Shortest Semantic Path Extraction (Heap-Optimized Dijkstra Algorithm over Sparse Semantic Graph)

For each topic cluster after community detection, this study performs shortest semantic path extraction over the sparse semantic graph using a heap-optimized version of Dijkstra's algorithm (Dijkstra, 1959).
This method efficiently extracts representative keyword sequences to guide the BART model's generation.

To avoid the computational explosion ($O(V^2)$) associated with dense semantic graphs, a sparsification strategy is employed:

- Each keyword retains only the top-k most semantically similar connections (e.g., k=5 or 10),
- Or applies a similarity threshold to filter edges,
- Thus controlling the number of edges E to be close to the number of vertices V ($E \approx V$).

On the constructed sparse graph, the heap-optimized Dijkstra algorithm further searches for the shortest semantic connection paths within each topic cluster. Compared to traditional array-based Dijkstra ($O(V^2)$), the heap-optimized version achieves a complexity of $O(E + V \log V)$, significantly improving computational efficiency.

Overall Process:

1. **Sparse Graph Construction:**
   Retain only the most semantically significant edges for each keyword to form a near-sparse semantic network.
2. **Shortest Path Search:**
   Use heap-optimized Dijkstra algorithm to find the shortest paths, which are extracted as structured prompts for BART generation.

This strategy not only reduces computational resource consumption but also preserves semantic coherence and thematic focus, ensuring the quality and scalability of the generated summaries.

3.6 Structure-Guided Generation (BART Model)

At the stage of shortest semantic path extraction, the BDS workflow extracts semantically coherent and structurally focused keyword sequences for each topic cluster.
These keywords not only effectively cover the core concepts of each topic but also help reduce information redundancy and mitigate semantic drift.

During the generation phase, the extracted keyword sequences, combined with the original text summaries, are fed into the BART model (Lewis et al., 2019) as structured prompts.
Specifically, the keyword sequences are serialized to highlight their semantic connections within each topic community, serving as thematic focal points during text generation.

This structure-guided strategy significantly mitigates issues such as semantic drift and thematic diffusion often observed in unguided BART generation, enhancing the quality of the generated summaries in the following three dimensions:

- **Thematic Focus:**
  The generated content revolves around the specified prompts, maintaining clear topic boundaries.
- **Semantic Coherence:**
  The relationships between keywords maintain natural and coherent semantic transitions.
- **Readability:**
  The generated text possesses a clear logical structure and improved overall readability.

In summary, by integrating sparse semantic graph construction, precise path extraction, and structure-guided generation, the proposed BART-Dijkstra-SA workflow not only achieves deep semantic modeling and computational efficiency but also demonstrates high scalability and application potential in large-scale text processing scenarios.

4. Conclusion

In this study, we proposed the BART-Dijkstra-SA (BDS) workflow as a scalable and systematic solution for semantic summarization and topic structuring of large-scale text data.
By integrating simulated annealing-based topic number optimization, LDA-based topic modeling, TF-IDF and cosine similarity-based semantic graph construction, Leiden community detection, and heap-optimized Dijkstra shortest path extraction, the BDS workflow successfully addresses key challenges in thematic focus, semantic consistency, and computational scalability that conventional models encounter when processing large-scale text corpora.

Our experiments demonstrate that leveraging sparse semantic graphs combined with structure-guided prompting significantly improves the quality of BART-generated summaries, ensuring clearer thematic boundaries, stronger semantic coherence, and higher readability.
Moreover, the BDS workflow's modular architecture and controlled time complexity make it highly adaptable to a wide range of applications, including misinformation analysis, news monitoring, large-scale knowledge management, and advanced NLP-driven decision support systems.

Future research could explore several directions to enhance the BDS framework:
First, developing more advanced semantic sparsification strategies could further reduce graph construction and search costs.
Second, integrating dynamic prompt optimization techniques, such as reinforcement learning or graph neural networks (GNNs), could adaptively refine keyword selection to maximize generative performance.
Third, extending the BDS framework to multilingual or domain-specific corpora would validate its generalizability across different linguistic and contextual settings.

Overall, this study contributes a novel, scalable workflow that bridges structural modeling, heuristic optimization, and neural text generation, providing a solid foundation for future advancements in natural language understanding and generation technologies.

5.References

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3, 993–1022.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of

communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10), P10008. https://doi.org/10.1088/1742-5468/2008/10/P10008

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1), 269–271. https://doi.org/10.1007/BF01386390

Hasan, M., Rahman, A., Razaul, M., Khan, M., & Islam, M. (2020). Normalized approach to find optimal number of topics in Latent Dirichlet Allocation (LDA). *International Journal of Advanced Computer Science and Applications*, 11(3), 405–411.

Kannan, S., Karuppusamy, S., Nedunchezhian, A., Venkateshan, P., Wang, P., Bojja, N., & Kejariwal, A. (2016). Big data analytics for social media. In *Big Data* (pp. 63–94). Academic Press. https://doi.org/10.1016/B978-0-12-805394-2.00003-9

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598), 671–680. https://doi.org/10.1126/science.220.4598.671

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2019). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*. https://arxiv.org/abs/1910.13461

Pathik, N., & Shukla, P. (2020). Simulated annealing-based algorithm for tuning LDA hyperparameters. In *Soft Computing: Theories and Applications* (pp. 469–479). Springer. https://doi.org/10.1007/978-981-15-4032-5_47

Reuters Institute for the Study of Journalism. (2023). *Digital News Report 2023*. Reuters Institute. https://reutersinstitute.politics.ox.ac.uk/digital-news-report/2023

Silverman, C. (2016). This analysis shows how viral fake election news stories outperformed real news on Facebook. *BuzzFeed News*. https://www.buzzfeednews.com/article/craigsilverman/viral-fake-election-news-outperformed-real-news-on-facebook

Traag, V. A., Waltman, L., & van Eck, N. J. (2019). From Louvain to Leiden: Guaranteeing well-connected communities. *Scientific Reports*, 9(1), 5233. https://doi.org/10.1038/s41598-019-41695-z

Zhang, W., Yoshida, T., & Tang, X. (2011). A comparative study of TF*IDF, LSI, and

multi-words for text classification. *Expert Systems with Applications*, 38(3), 2758–2765. https://doi.org/10.1016/j.eswa.2010.08.066