Text Similarity from Time Series Perspective

高级商务智能　期末项目

姓名: 范维

学号: 2023214429

班级: 信息管理与商务智能

目录

Text Similarity from Time Series Perspective

范维

2024 年 6 月 9 日

1 Abstract

This paper focuses on the task of measuring text similarity.[1] Traditional methods for text similarity, such as the Bag-of-Words model, rarely consider the order of words in the text, which can lead to inaccurate similarity measures. Our approach transforms text information into word or sentence representation vectors using models like Word2Vec, and then views these representation tensors from a time series perspective. By employing Dynamic Time Warping (DTW) [1], we calculate the distance between texts and convert this distance into a similarity measure. This method takes into full account the order of words and sentences, providing a more accurate measurement of text similarity. We designed unsupervised experiments to measure similarity and compared our results with expert human ratings across various dimensions. Our model demonstrated promising performance. In the future, we may improve the text embedding model and develop a supervised prediction model to further enhance prediction accuracy.

[1]The code is hosted in the library `https://github.com/WinstonFanWei/TextSimilarity`.

2 Introduction

The accurate measurement of text similarity is a critical task in Natural Language Processing (NLP) with applications spanning information retrieval, document clustering, plagiarism detection, and semantic search. Traditional methods for text similarity, such as the Bag-of-Words (BoW) model, are limited by their disregard for the sequential nature of text, which often leads to inaccurate similarity assessments.

Most text similarity measurements convert two texts into highly summarized vectors using models, then calculate their similarity by computing the cosine similarity of these vectors. The Bag-of-Words (BoW) model simplifies text to a collection of word frequencies, ignoring word order and context. Models like TF-IDF and LDA, which are based on the BoW model, share the same issue.

Although models such as Word2Vec [2] consider word order when generating word embedding vectors, they do not fully utilize global word order information during the final comparison of document similarities. The Word Mover's Distance (WMD) model improves similarity calculations by aligning words in the texts, but it also overlooks word order.

In the time series perspective, we already have methods to measure similarity. By representing text as a time series and applying these methods, we can account for word order information, an approach that has been largely overlooked.

This modeling approach allows us to represent text as a matrix of word vectors, directly feeding the lowest-level text representation into the model, potentially enhancing its performance. Unlike structural methods in NLP that require assumptions about and learning the parameters of the generative process, viewing text as a time series enables similarity calculations through alignment without these assumptions.

Therefore, our alignment approach has many advantages over the traditional structural approach, which summarizes document semantics into a vector by training generative language models.

To address these limitations, we propose an innovative approach that leverages advanced text representation techniques and sequence alignment methods. Specifically, we transform text information into word or sentence representation vectors using models like Word2Vec. These vectors are then analyzed from a time series perspective, allowing us to apply Dynamic Time Warping (DTW) [1] to measure the distance between texts. This method preserves the order of words and sentences, thereby providing a more accurate measurement of text similarity.

Our experimental setup involves a comprehensive analysis using a legal document dataset[2], consisting of 144 legal documents and an accompanying Excel file with 100 expert-annotated similarity scores. The experiments were designed to compare the performance of different similarity measurement methods:

[2]The same dataset as that metioned in [3].

1. Method 1: We used TF-IDF [4] to create a BoW model and computed similarity using cosine similarity.

2. Method 2: Each document was modeled using a pre-trained Latent Dirichlet Allocation (LDA) [5] model to obtain topic distribution vectors. Document similarity was then computed using cosine similarity on these vectors.

3. Method 3: At the word or sentence level, documents were modeled using LDA. We measured topic distance by considering the distribution of words or sentences within each topic using word movement distence (WMD) [6]. For each document, the topic of each token was determined, and DTW was used to calculate the distance between documents, effectively considering word order information.

To evaluate the effectiveness of these methods, we employed various metrics including Root Mean Squared Error (RMSE), F1-score, and correlation with human-annotated scores. Our results demonstrate that the DTW-based method, which incorporates word order information, significantly outperforms traditional methods in aligning with expert human ratings.

This paper makes several contributions to the field of text similarity measurement. First, we introduce a novel application of DTW to text similarity, which effectively incorporates the sequence of words. Second, we provide empirical evidence through unsupervised experiments that our model aligns well with human judgments across multiple dimensions.

Looking forward, we plan to enhance our approach by refining the text embedding models and developing supervised prediction models to further improve prediction accuracy. This could involve integrating advanced contextualized embeddings, such as those produced by BERT [7], and exploring other alignment techniques that better capture the nuances of textual data.

3 Related Work

In the realm of text similarity measurement, traditional approaches have been extensively surveyed by (H.Gomaa & A. Fahmy, 2013)[8]. They categorize these approaches into three main types: String-Based, Corpus-Based, and Knowledge-Based methods. String-Based methods focus on character and term matching, such as the Longest Common Substring and Cosine Similarity. Corpus-Based methods utilize large text corpora to derive semantic relationships, exemplified by Latent Semantic Analysis (LSA) and Explicit Semantic Analysis (ESA). Knowledge-Based methods leverage structured semantic networks like WordNet to assess similarity based on shared meanings and relationships. This comprehensive survey underscores the limitations of conventional methods in capturing the sequential nature of text, which our proposed time series perspective on a Corpus-Based model aims to address.

(Gong et al., 2018)[9] presents a novel approach to matching long documents with their summaries by utilizing hidden topic modeling and domain-specific word embeddings. This method addresses the challenges of vocabulary and context mismatches that arise due to the varying lengths of the texts being compared. The authors propose a multi-view generalization of the document through hidden topics to bridge the gap between detailed texts and their abstractions.

(Kusner et al., 2015)[6] introduces the Word Mover's Distance (WMD), a novel metric for measuring the distance between text documents based on word embeddings. The WMD represents documents as weighted point clouds of embedded words and calculates the minimum cumulative distance that words from one document need to "travel" to match the point cloud of another document. This approach leverages the semantic meaning captured by word embeddings like Word2Vec, which can capture relationships between words. The WMD is shown to be highly effective, outperforming several state-of-the-art document distance metrics in real-world document classification tasks.

(Matuschek et al., 2008)[1] This article introduces a novel approach for comparing typed texts by transforming them into artificial time series and then measuring their similarity using Dynamic Time Warping (DTW) distance. The method involves counting relevant keywords within a sliding window applied to the text to create the time series, which can then be analyzed with time series analysis techniques. The authors demonstrate that this technique is effective for recognizing similar texts, including those in different languages, and could aid in plagiarism detection and information retrieval. They also discuss the potential for further development, such as incorporating additional text preprocessing steps and combining this method with other information retrieval techniques.

4 Model

4.1 LDA and Word2Vec

Latent Dirichlet Allocation (LDA) [5] is a generative statistical model, used to identify latent topics within a collection of documents. As Probabilistic graphical model of LDA shows, LDA assumes each document is a mixture of several topics, with each topic being characterized by a distribution over words. By using Dirichlet distributions to model the variability of topic proportions per document and word distributions per topic, LDA can infer the hidden topic structure from the observed words. This inference typically involves techniques like Variational Bayes or Gibbs Sampling.

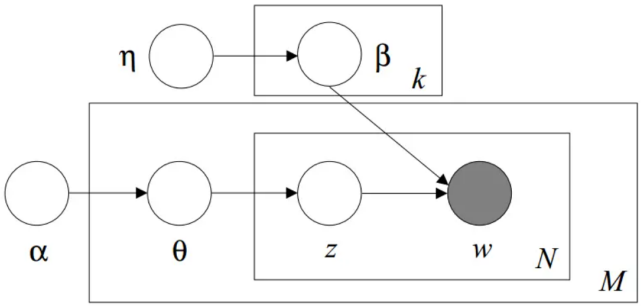

图 1: Probabilistic graphical model of LDA

We use LDA to model the probability distribution of words in each topic and the probability distribution of topics in each document, which can then be used in TS-DS and other models.

Word2Vec [2] is a popular unsupervised learning algorithm that learns word embeddings from a large corpus of text. As Word2Vec architectures shows, it is based on the skip-gram model, which predicts the context words given the target word, and the continuous bag-of-words model, which predicts the target word given the context words.

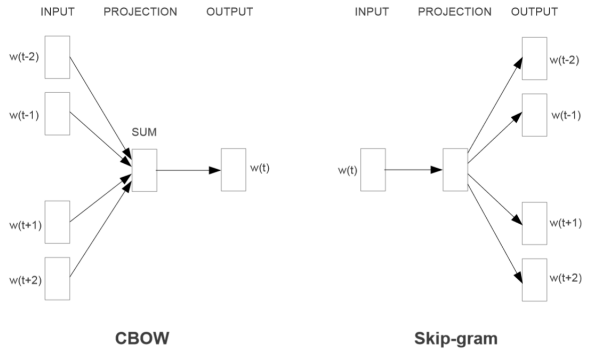

图 2: Word2Vec architectures

We use Word2Vec to generate word embeddings for each word in the corpus, which can then be used to represent the documents in the latent space.

4.2 Measure Document Similarity from Time Series Perspective(TS-DS)

As The entire TS-DS model illustrates, the TS-DS model first uses the word2vec model to embed every word that appears in the entire document collection. By calculating the distances between these word embeddings, we obtain the distances between words. Simultaneously, LDA is used to model the entire document collection, yielding the word distributions for each topic. Subsequently, using each topic's word distribution and the distances between words, we can calculate the Word Mover's Distance (WMD) between topics. Finally, we represent each document as a sequence of topics, where each position corresponds to the most probable topic for each word in the document. We then use Dynamic Time Warping (DTW) to calculate the distances between the topic sequences of each document, which is then converted into similarity.

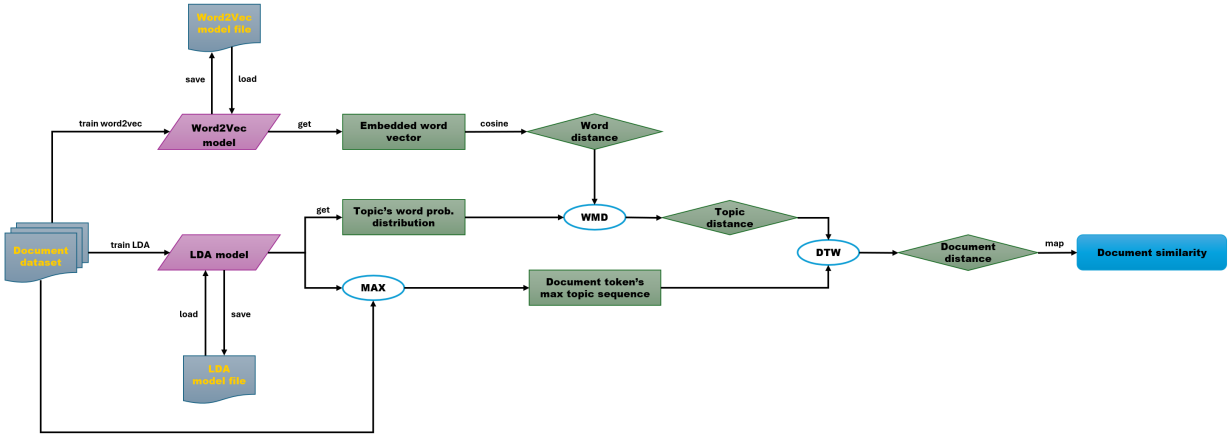

图 3: The entire TS-DS model

4.2.1 Topic distance matrix(TDM)

"Topic distance" is an intriguing way to measure the semantic and thematic divergence between topics within a corpus. It leverages the power of Word2Vec for generating dense vector representations of words, capturing their semantic relationships. Topic distance is calculated by first determining the distances between individual words using Word2Vec and cosine similarity, and then applying these distances to the word distributions of topics using WMD. This approach allows for a detailed and nuanced comparison of the thematic structures revealed by LDA, where each topic is characterized by a specific distribution of words.

To compute the topic distance, one begins by embedding words into a continuous vector space using Word2Vec, allowing for a nuanced understanding of word similarities. However, if we calculate the distances between all pairs of words, the computational load becomes enormous. Therefore, we consider taking measures to reduce the number of words in the corpus, which in turn reduces the cost of computing and storing word similarities. Currently, we use a filter to exclude words that appear less than five times. We then perform cosine comparisons on the remaining word pairs to obtain their similarities and then transfer them to distance by (4.2).

The crux of topic distance lies in comparing word distributions across topics. This is achieved by conceptualizing each topic as a probability distribution of words and calculating the distance between these distributions, through the Word Mover's Distance (WMD), as (4.4) and (4.5) show.

WMD operates by considering the minimum amount of "work" required to transform one distribution of topic vectors into another. This "work" is quantified as the cumulative distance words from one topic need to "travel" to match the distribution of words in another topic. Essentially, it measures how much one topic's word distribution must be altered to resemble another topic's word distribution. This method provides a nuanced measure of the thematic dissimilarity between topics, capturing subtle differences in word usage and context that might be overlooked by simpler metrics.

In essence, topic distance is a multi-faceted metric that amalgamates the semantic closeness of individual words with the topic composition of words. It offers a comprehensive assessment of topic divergence by considering both the distribution of words within topics and the semantic relatedness of those words. This approach not only enhances our ability to discern the thematic landscape of textual data but also provides a framework for analyzing topic similarity.

With parameters $Dim := Dimension\ of\ word\ embedding$, $N := Total\ word\ count$,

$$WordVec_i := (w_i^1, w_i^2, \ldots, w_i^{Dim}) \tag{4.1}$$

$$D_{word}^{i,j} = 1 - \frac{WordVec_i \times WordVec_j}{|WordVec_i| \times |WordVec_j|} \tag{4.2}$$

$$M := \{D_{word}^{i,j}\}_{N \times N} \tag{4.3}$$

For the topic distance matrix TDM, we use WMD to determine each element. For example, the distance between topic x and topic y is given as a optimal problem:

WMD optimal problem:

$$TDM_{x,y} = \min_{Q=\{q(1,1),q(1,2),\ldots,q(N,N)\}} \left(\sum_{i\in\{1,\ldots,N\}} \left(\sum_{j\in\{1,\ldots,N\}} q(i,j) \times M(i,j) \right) \right) \tag{4.4}$$

$$s.t. \begin{cases} \sum_{j\in\{1,\ldots,N\}} q(i,j) = w_x^i,\ \forall i \in \{1,\ldots,N\} \\ \sum_{i\in\{1,\ldots,N\}} q(i,j) = w_y^j,\ \forall j \in \{1,\ldots,N\} \end{cases} \tag{4.5}$$

With parameter $E := Total\ topic\ count$,

$$TDM = \{TDM_{i,j}\}_{E\times E} = \begin{pmatrix} 0 & TDM_{1,2} & TDM_{1,3} & \ldots \\ TDM_{1,2} & 0 & TDM_{2,3} & \ldots \\ TDM_{1,3} & TDM_{2,3} & 0 & \ldots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}_{E\times E} \tag{4.6}$$

We can see from (4.6) that the TDM matrix is a symmetric matrix with diagonal elements equal to 0. This is because, unlike the calculation of KL divergence [10], the transportation distance from topic x to topic y is equal to the transportation distance from topic y to topic x in WMD. Additionally, if the word probability distributions of the two topics are equal, then the $q(i,j)$ term in (4.4) will always be 0, thus all diagonal elements are 0.

4.2.2 Documents distance from topic series perspective using DTW

We extract the most likely topic for each word in every document from the LDA model, forming a sequence of topics for each document based on the word order, which is used in (4.9). By leveraging the distances between topics that we obtained previously, we can employ the DTW method to compute the distance between documents.

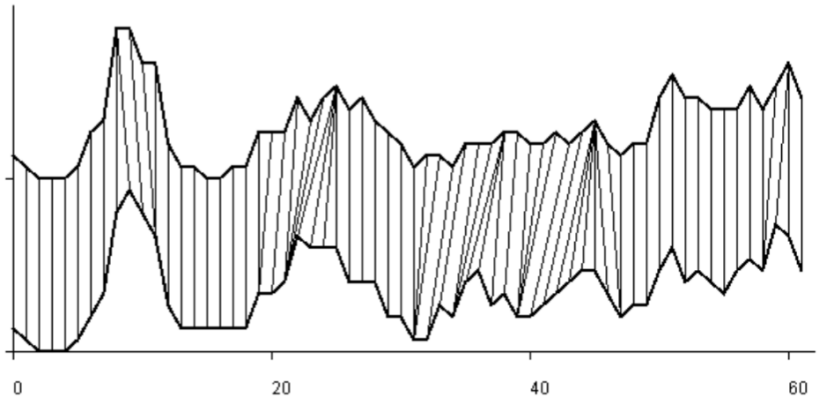

图 4: Dynamic Time Warping (DTW)

Dynamic Time Warping (DTW) [1] is a well-known algorithm for measuring similarity between two sequences, which may vary in time or speed. It is particularly effective for sequences that are misaligned in the time dimension. DTW works by finding an optimal alignment between two sequences by minimizing the cumulative distance. The algorithm uses a distance matrix, which is TDM in our model, where each element (i, j) represents the distance between the $i-th$ element of the first sequence and the $j-th$ element of the second sequence. The goal is to find a path through this matrix that minimizes the total distance while respecting the ordering of the elements. This path, known as the warping path, allows for stretching and compressing of the sequences to achieve the best alignment. The DTW distance is then the sum of the distances along this optimal path.

With parameter $S_1 := Sequence\ length\ of\ Doc1,\ S_2 := Sequence\ length\ of\ Doc2$,

---

DTW optimal problem:

$$DTW = \min \sum_{k=1}^{K} l_k \tag{4.7}$$

$$s.t. \begin{cases} l_1 = (1,1) \\ l_K = (S_1, S_2) \\ Given\ l_k = (s_1, s_2),\ then\ l_{k+1} = (s_1', s_2')\ with\ 0 \leqslant s_1' - s_1 \leqslant 1\ and\ 0 \leqslant s_2' - s_2 \leqslant 1. \end{cases} \tag{4.8}$$

$$where\ (s_1, s_2) := distance(s_1, s_2) = TDM(TOPIC_{Doc1\ on\ s_1}, TOPIC_{Doc2\ on\ s_2}) \tag{4.9}$$

DTW is the sum of the distances between the corresponding elements of the two sequences. However, because DTW is not normalized, it increases with document length, even though document similarity is not necessarily related to document length. Therefore, we need to normalize DTW. There are some ways to do this, for instance, $DTW_{standard} = \frac{\sqrt{DTW}}{K}$. We now use (4.10) to normalize DTW.

$$DTW_{standard} = \frac{DTW}{K} \tag{4.10}$$

So, for the final document distance calculation, we use DTW to process the topic sequence information of the documents. This approach allows us to consider the overall word order, thereby making the calculation of document similarity more accurate.

4.2.3 Map document distance to similarity

Since the distance between documents is a value in $[0, +\infty)$, while document similarity is a value in $[0, 1]$, we need to map the distance to similarity. In previous research, there is no standardized method for converting document distance to similarity, so we have developed our own approach.

Obviously, all normalized DTW distances are less than the maximum entry in the $TDM\ matrix$, which we can observe once the $TDM\ matrix$ is obtained. Based on this observation, we narrow the range of document distances to be $[0, \max_{i,j}\{TDM_{i,j}\})$.

Therefore, we can design an activation function to perform this mapping. For example, taking both linear and nonlinear relationships into consideration, the similarity between document A and document B can be calculated as follows:

$$Sim_{A,B} = -\alpha \times DTW^{A,B}_{standard} + 1 + e^{-\beta \times DTW^{A,B}_{standard}} \tag{4.11}$$

Finally, when new documents come, we can use the whole process mentioned in chapter 4.2 to calculate the similarity between them.

5 Experiments

5.1 Data preprocessing

We use a legal document dataset, consisting of 144 legal documents and an accompanying Excel file with 100 expert-annotated similarity scores. We first read in the text documents. And then we used NLTK for tokenization, stop word removal, and stemming. Since sentence-level data is also required for comparison experiments, we used Spacy to perform sentence segmentation on the texts. All the processing results are stored, indexed by the file names.

5.2 Set up

5.2.1 Model training

The training process for our model involved several key steps to ensure the accuracy and effectiveness of our document similarity measurements. Initially, we constructed a dictionary from the training data, which included reading the text content of each document and storing it in a list. This list was then converted into a bag-of-words model using the dictionary. To capture the thematic structure of the documents, we trained an LDA model. If the model had not been preloaded, we trained it with 20 topics over 40 passes, ensuring robust topic modeling. The model was saved for future use. Additionally, we trained a Word2Vec model on the same text data to generate word embeddings, which capture semantic relationships between words. This training was conducted with a vector size of 50, a window size of 10, and other specified parameters. Both the LDA and Word2Vec models were trained using a fixed random seed to ensure reproducibility, and the training times for each model were recorded.

5.2.2 Obtaining Topics

1. Token-Level Topic Probabilities: For each document, we processed the text content to determine the topic probabilities for each token. Initially, each token was converted into its bag-of-words (BOW) representation using the previously constructed dictionary. We then applied the trained LDA model to obtain the topic distribution for each token. This distribution included probabilities for all topics, ensuring comprehensive topic coverage. The results were simplified by storing only the probabilities of each topic for every token and identifying the most probable topic for each token.

2. Sentence-Level Topic Probabilities: For sentence-level analysis, we adopted two methods to determine the topic probabilities: averaging the token topic distributions within a sentence or directly obtaining the topic distribution for the entire sentence. In our current approach, we used the second method, which involved applying the LDA model to get the topic distribution for each sentence. This method provided a straightforward way to capture the dominant topic for each sentence by identifying the topic with the highest probability.

All processing results were stored in a structured format, indexed by document identifiers, facilitating efficient retrieval and further analysis.

5.2.3 Methods of measuring document similarity

In this study, we employed several methods to calculate document similarity, with a primary focus on Dynamic Time Warping (DTW). The process involves computing a similarity score between two documents by leveraging their topic distributions at either the token or sentence level.

1. TS-DS on token level: This method is our PRIMARY FOCUS. We used the DTW algorithm to calculate the similarity score between two documents through the token-level topic distributions.

2. TS-DS on sentence level: We used the DTW algorithm to calculate the similarity score between two documents through the sentence-level topic distributions. We obtain the sentence-level topic distributions by giving the LDA model the sentence as a ducument. The model will return the topic distribution of the sentence. In this method, we also use TS-DS as method 1.

3. TF-IDF and cosine similarity: We utilize the TF-IDF model to characterize the entire dataset, obtaining representations for each document. Subsequently, we employ cosine similarity to compute the similarity between every pair of documents.

4. Document topic distribution similarity: We employ the LDA model to model the entire dataset, obtaining topic distribution representations for each document. Subsequently, we utilize cosine similarity to calculate the similarity between every pair of documents.

5.3 Evaluation

We conduct experiments using the four models mentioned above, recording all obtained similarities. Subsequently, we evaluate the accuracy of each method by comparing the obtained similarities with those provided by experts using the following three metrics (5.1), (5.2), (5.3).

1. RMSE:

$$RMSE = \sqrt{\frac{1}{100}\sum_{i=1}^{100}(Sim_i - Sim_i^*)^2} \tag{5.1}$$

2. F1-score:

$$F1 - score = \frac{2}{\frac{1}{P} + \frac{1}{R}} \tag{5.2}$$

   P is the precision and R is the recall in classification problem.

   For both predicted and expert-provided similarities, we convert them into binary variables: similarities greater than or equal to 0.5 are represented as 1, indicating similarity, while those less than 0.5 are represented as 1, indicating dissimilarity, following the approach outlined in [3].

3. Correlation:

$$Correlation = \frac{COV(Sim, Sim^*)}{std(Sim) \times std(Sim^*)} \tag{5.3}$$

表 1: The effects of all methods

Method	RMSE	F1-score	Correlation
TS-DS on token level	0.3253	0.7156	0.5619
TS-DS on sentence level	0.2506	0.7551	0.5442
TF-IDF and cosine similarity	0.2185	0.6923	0.7056
Document topic distribution similarity	0.3448	0.8224	0.6219

The effects of all methods are showed in table The effects of all methods, To enhance authenticity, you can see the result in Figure 5 which recorded the experimental results in log file with time.

Although the performance of TS-DS on the token level did not surpass the document topic distribution similarity model, it exhibited similar effectiveness to the baseline model and outperformed the other two models. This is acceptable considering that the model is still under development. There are numerous aspects in which this model can be further improved, which we will elaborate on in the Future Work section.

```
2024-05-25 10:15:23,811 - Utils - DEBUG
---------------------- RMSE ----------------------
[0, +inf] RMSE smaller is better.
[mySimilarity] RMSE: 0.3253
[Similarity_cosine] RMSE: 0.2185
[Similarity_doc_topic] RMSE: 0.3448
[Similarity_sentence_topic] RMSE:  0.2506
--------------------------------------------------

2024-05-25 10:15:23,812 - Utils - DEBUG
---------------------- Correlation ----------------------
[-1, 1] Correlation bigger is better.
[mySimilarity] CORR: 0.5619
[Similarity_cosine] CORR: 0.7056
[Similarity_doc_topic] CORR: 0.6219
[Similarity_sentence_topic] CORR:  0.5442
-------------------------------------------------------

2024-05-25 10:15:23,815 - Utils - DEBUG
---------------------- F1-score ----------------------
[0, 1] F1-score bigger is better.
[mySimilarity] F1-score: 0.7156
[Similarity_cosine] F1-score: 0.6923
[Similarity_doc_topic] F1-score: 0.8224
[Similarity_sentence_topic] F1-score:  0.7551
--------------------------------------------------
```

图 5: Result

6 Conclusion

The cornerstone of this research is the introduction of an innovative approach to measuring text similarity that fully embraces the sequential nature of text information. Traditional methods in the domain of Natural Language Processing (NLP), such as the Bag-of-Words (BoW) model, have frequently neglected the importance of word order, resulting in less accurate similarity assessments. Even advanced models like Word2Vec, while capturing local context and some order information through its predictive framework, often fall short of considering the global sequence of words across entire documents.

Our methodology leverages text representation techniques like Word2Vec to transform text into word or sentence representation vectors. And We employ Latent Dirichlet Allocation (LDA) to uncover the latent topics within the document collection, This topic modeling provides a structured way to consider the content of texts as series.

Building upon the word embeddings and topic distributions, we introduce Word Mover's Distance (WMD) to quantify the thematic divergence between topics. WMD creatively represents topics as probability distributions of words and calculates the minimum work required to transform one distribution into another, thus providing a measure of semantic and thematic distance that respects the order of words.

By viewing these vectors through a time series perspective and applying Dynamic Time Warping (DTW), we have developed a method that preserves the order of words and sentences, thereby providing a more nuanced and precise measurement of text similarity.

The results of our unsupervised experiments, conducted on a legal document dataset with expert-annotated similarity scores, demonstrate the effectiveness of our model. Our approach, which integrates the strengths of Word2Vec, WMD, and LDA with the alignment capabilities of DTW, shows a strong correlation with human judgments and outperforms traditional methods that do not account for the sequential nature of text.

The innovative aspect of our approach is the application of DTW to text similarity, which effectively incorporates the sequence of words. This offers a more comprehensive understanding of the thematic and semantic relationships within texts.

In conclusion, by developing a methodology that combines the local insights of Word2Vec with the global perspective of WMD and LDA, followed by the sequence alignment of DTW, we have created a comprehensive framework that captures the intricate order and arrangement of words within texts. This innovation is poised to enhance text analysis across a variety of NLP applications, offering a more accurate and nuanced understanding of text similarity.

7 Future Work

As we look towards enhancing the accuracy and efficiency of our text similarity measurement model, several promising avenues for future research and development present themselves:

Latent LSTM Allocation [11]: We aim to explore the integration of Latent LSTM Allocation to improve the accuracy of our current topic modeling. This method may offer a more nuanced understanding of the temporal dynamics within documents, capturing the evolution of topics over time.

Advanced Topic Sequence Extraction: Rather than simply taking the most probable topic for each word to extract sequence information, we can apply a more sophisticated approach that utilizes WMD to dynamically calculate distances between sequences at each position. Although computationally intensive, this method could preserve more of the semantic information within the documents.

Leveraging Large Language Models for Embeddings: The advent of large pre-trained language models like BERT provides an opportunity to obtain more contextually rich word embeddings. We plan to experiment with these models to calculate word distances, potentially leading to a more accurate representation of semantic relationships.

Custom DTW Model for Document Sequences: To better align with our model's needs, we intend to design a specialized version of the DT algorithm that is tailored for comparing document sequences. This could involve adapting the cost function to account for the unique characteristics of textual data.

Supervised Learning Approach: We are considering the development of a supervised learning model that can be trained on a labeled dataset to fine-tune our similarity measurement parameters. By segmenting our data and employing techniques such as cross-validation, we can iteratively improve our model's predictive accuracy.

Refining DTW Distance to Similarity Mapping: The method of converting DTW distances into similarity scores warrants further investigation. We plan to experiment with various activation functions and hyperparameter tuning to optimize this conversion process.

Application to Diverse Datasets: To strengthen the validation of our model, we will apply it to a wider range of datasets. This will not only test the model's versatility but also help refine our approach based on diverse textual content.

Performance Optimization: Given the computational complexity of our approach, particularly with the introduction of WMD in sequence extraction and the potential use of large language models, performance optimization will be a critical focus. This may involve algorithmic enhancements and the exploration of parallel processing techniques.

Integration with Information Retrieval Systems: Finally, we aim to integrate our text similarity model with information retrieval systems, enhancing their ability to deliver more relevant search results and improve user experience.

By pursuing these avenues of research, we are confident that our text similarity measurement model will become an even more powerful tool for applications across Natural Language Processing.

参考文献

[1] Matuschek, M., Schlüter, T., & Conrad, S. (2008). Measuring text similarity with dynamic time warping. Proceedings of the 2008 International Symposium on Database Engineering & Applications - IDEAS ' 08, 263. https://doi.org/10.1145/1451940.1451977

[2] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space (arXiv:1301.3781). arXiv. https://doi.org/10.48550/arXiv.1301.3781

[3] Bhattacharya, P., Ghosh, K., Pal, A., & Ghosh, S. (2022). Legal case document similarity: You need both network and text. Information Processing & Management, 59(6), 103069. https://doi.org/10.1016/j.ipm.2022.103069

[4] Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. Communications of the ACM, 18(11), 613–620. https://doi.org/10.1145/361219.361220

[5] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. Journal of machine Learning research, 3(Jan):993–1022.

[6] Kusner, M., Sun, Y., Kolkin, N., & Weinberger, K. (2015). From Word Embeddings To Document Distances. Proceedings of the 32nd International Conference on Machine Learning, 957–966. https://proceedings.mlr.press/v37/kusnerb15.html

[7] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (n.d.). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

[8] H.Gomaa, W., & A. Fahmy, A. (2013). A Survey of Text Similarity Approaches. International Journal of Computer Applications, 68(13), 13–18. https://doi.org/10.5120/11638-7118

[9] Gong, H., Sakakini, T., Bhat, S., & Xiong, J. (2018). Document Similarity for Texts of Varying Lengths via Hidden Topics. In I. Gurevych & Y. Miyao (Eds.), Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 2341–2351). Association for Computational Linguistics. https://doi.org/10.18653/v1/P18-1218

[10] Zhu, H. (n.d.). On Information and Sufficiency.

[11] Zaheer, M., Ahmed, A., & Smola, A. J. (2017). Latent LSTM Allocation: Joint Clustering and Non-Linear Dynamic Modeling of Sequence Data. Proceedings of the 34th International Conference on Machine Learning, 3967–3976. https://proceedings.mlr.press/v70/zaheer17a.html

[12] Deshpande, A., Jimenez, C. E., Chen, H., Murahari, V., Graf, V., Rajpurohit, T., Kalyan, A., Chen, D., & Narasimhan, K. (2023). C-STS: Conditional Semantic Textual Similarity (arXiv:2305.15093). arXiv. https://doi.org/10.48550/arXiv.2305.15093

[13] Al-Rfo+u, R., Perozzi, B., & Skiena, S. (2014). Polyglot: Distributed Word Representations for Multilingual NLP (arXiv:1307.1662). arXiv. https://doi.org/10.48550/arXiv.1307.1662