Electronic Research Archive

ERA, 31(6): 3609–3624.
DOI: 10.3934/era.2023183
Received: 14 January 2023
Revised: 13 April 2023
Accepted: 17 April 2023
Published: 24 April 2023

http://www.aimspress.com/journal/ERA

Research article

Sentence coherence evaluation based on neural network and textual features for official documents

Yunmei Shi[1], Yuanhua Li[2,*] and Ning Li[1]

[1] School of Computer, Beijing Information Science and Technology University, Beijing 100101, China
[2] Beijing Key Laboratory of Internet Culture and Digital Dissemination Research, Beijing 100101, China

*** Correspondence:** Email: liyuanhua7696@163.com.

Abstract: Sentence coherence is an essential foundation for discourse coherence in natural language processing, as it plays a vital role in enhancing language expression, text readability, and improving the quality of written documents. With the development of e-government, automatic generation of official documents can significantly reduce the writing burden of government agencies. To ensure that the automatically generated official documents are coherent, we propose a sentence coherence evaluation model integrating repetitive words features, which introduces repetitive words features with neural network-based approach for the first time. Experiments were conducted on official documents dataset and THUCNews public dataset, our method has achieved an averaged 3.8% improvement in accuracy indicator compared to past research, reaching a 96.2% accuracy rate. This result is significantly better than the previous best method, proving the superiority of our approach in solving this problem.

Keywords: sentence coherence; XL-Net; repetitive words; feature fusion

1. Introduction

Sentence coherence refers to the smooth connection and natural transition between sentences, which is an important foundation for coherent discourse in various natural language processing tasks such as automated essay scoring [1], writing quality assessment [2], and machine translation [3]. There are two main aspects of sentence coherence: the first is semantic, including the consistency of the content between sentences and the coherence of the text around a unified theme; the second is formal,

including the consistency of sentence structure, appropriate words order, and coherence between adjacent sentences.

1. Coherent text

(1) In order to prevent and control environmental noise pollution, protect and improve the living environment, this method is formulated based on the Environmental Noise Pollution Prevention and Control Law and in combination with the actual situation of this city. (2) This method applies to the prevention and control of environmental noise pollution within the administrative area of this city.

2. Incoherent text

(1) In order to prevent and control environmental noise pollution, protect and improve the living environment, this method is formulated based on the Environmental Noise Pollution Prevention and Control Law and in combination with the actual situation of this city. (2) Restaurants and entertainment venues that may cause noise pollution are not allowed to be set up in residential buildings.

Figure 1. Examples of coherent and incoherent text.

Figure 1 provides examples of coherent and incoherent text. In Example 1, the two sentences are semantically connected, and the content is logically coherent. In contrast, Example 2 lacks semantic connections and logical relationships between the two sentences.

With the continuous development and improvement of intelligent e-government, the automatic generation of official documents can greatly reduce the workload of document writers. However, whether using extractive generation methods [4,5] or generative generation methods [6,7], it is necessary to evaluate the coherence of sentences in the text to ensure that the generated document content has good coherence and readability, and to improve the quality of the text. Previously, some researchers tried to maintain sentence coherence by defining and using features to capture logical and dependency relationships between sentences [8]. It requires a lot of time for feature engineering and is inefficient. With the increase in sentence length and complexity, the noise that interferes with sentence coherence also increases, leading to a significant decrease in the accuracy of this type of method. With the rapid development and excellent performance of deep learning technology in the field of NLP, researchers have begun to use neural network-based methods to explore the deep semantic information of sentences and capture semantic logical relationships between sentences [9,10], achieving higher accuracy in sentence coherence evaluation tasks. However, neural network-based methods lack consideration of text features, which have a significant promoting effect on improving sentence coherence and text readability.

Currently, there is no research on sentence coherence evaluation specifically for the domain of official documents. Official documents can be categorized into 15 genres, among which Regulations and Measures documents exhibit two distinct characteristics. First, the content of these documents is broad, the sentences are lengthy, and the sentence structures are complex. Second, adjacent sentences often contain repetitive words. The first characteristic makes this task unsuitable for feature-based methods. Although neural network-based methods are proficient in capturing semantic and logical information between long sentences, they often overlook the second characteristic mentioned above.

In this paper, we address the problem of lacking sentence coherence in the autogeneration of Regulations and Measures documents by conducting a research on sentence coherence evaluation specific to official documents. Based on the characteristics of official documents content, our unique contribution is that we introduce the repetitive words features into sentence coherence evaluation for the first time. We combine neural networks with the repetitive words features to evaluate sentence coherence by overcoming the shortcomings of traditional neural network-based methods. We conduct experiments on official documents dataset and THUCNews public dataset. The experimental results demonstrate that our model has significantly higher accuracy compared to contrast models.

2. Related work

Sentence coherence is an important aspect of text quality assessment and a key factor in ensuring text readability. Early research on sentence coherence evaluation used feature-based methods, including entities features, rhetorical relationships, and grammatical structures, with entities features being the most representative. Later, more and more researchers began to explore how to use neural networks to solve the problem of sentence coherence evaluation. Among them, one type of method uses vector operations to enhance the model's ability to capture semantic and logical relationships between sentences, while another integrates text features into sentence-level vector representations. This chapter will introduce relevant research on coherence evaluation in order, from entity-driven methods to neural network-based methods, and then introduce the related technologies applied in this paper.

2.1. Entity-Driven methods

Entities are objective and distinguishable objects that mainly appear in noun form in text. Entity-driven methods mainly measure the coherence by leveraging the distribution of entities in the text. Barzilay et al. [11] first introduced the concept of entities into coherence evaluation and proposed a new framework for representing and measuring coherence the Entity Grid Model (EGM). The authors abstract text as a set of entity grids, where rows correspond to sentences and columns correspond to entities, to simulate the distribution of entities in the text. Although EGM considers relatively simple and limited features, it has inspired subsequent research. Louis et al. [12] built upon EGM and used Hidden Markov Model (HMM) to learn syntactic features of text, but this method requires significant time to analyze and statistically process sentence constituents, leading to lower overall efficiency. Strube et al. [13] represented scientific papers as graphs, where entity nodes can only be connected to sentence nodes (only when entities appear in sentences), and connections are established when two sentences share the same entities. This method is suitable for evaluating coherence in scientific paper abstracts, but not for open-domain text.

Overall, entity-driven methods mainly indirectly approximate the potential semantic information and logical relationships between sentences by extracting the distribution of entities in the text, which has limitations.

2.2. Neural network-based methods

Li et al. [14] were the first to propose using Recurrent Neural Networks (RNN) for automatically learn sentence semantic vector representations, and then concatenate the two sentence vectors to

calculate coherence scores. This approach eliminates the need for tedious feature engineering and enhances the discriminative ability of the model by automatically learning the semantic feature differences between positive and negative samples. Mou et al. [15] used Long Short-term Memory Networks (LSTM) to obtain distributed vector representations of adjacent sentences, and evaluated sentence coherence by combining the features of sentence vectors using dot-product and dot-difference operations. Based on this method, Luan et al. [16] introduced a static attention mechanism to enhance the model's ability to capture semantic logical relationships between sentences. Xu et al. [17] used simple GloVe word vectors to obtain average feature vector representations of sentences, and then computed the cascading features of sentences using dot addition and dot product operations before inputting them into a neural network to evaluate their coherence. These methods use vector operations to obtain combined features of sentence vectors, enhancing the model's ability to learn semantic logical relationships between sentences, but they still lack consideration of the important role of text features in sentence coherence.

Some researchers used vector operations to enhance feature representation. Xu et al. [18] first summed all entity feature vectors appearing in the sentence, and then used dot-product operation to fuse the summation result into the sentence feature vector. Du et al. [19] first extracted entities from the sentence, obtained distributed representations of sentences and entities using LSTM, and then fused the entities features into the sentence vector using dot-addition and dot-product operations. Liu et al. [20] directly concatenated entities features vectors at the end of the sentence vector as markers to retain more original sentence information. These methods effectively fuse entities features into sentence-level vectors through simple methods to enhance the neural network's ability to learn entity linking relationships and capture sentence logical information.

2.3. Methods proposed in this paper

For official documents, entities exhibit large variations in quantity, types, and distribution across adjacent sentences, making entities features less distinctive. Conversely, repetitive words have a relatively stable semantic relationship and are easier to learn features across adjacent sentences. Therefore, we propose to introduce the repetitive words features into the study of coherence evaluation for the first time, using a combination of neural networks and repetitive words features to assess coherence between sentences. This overcomes the drawback of feature-based methods that rely too much on feature extraction and engineering, while also addressing the limitation of neural network-based methods that lack effective integration of coherence features.

The research framework diagram for this paper is shown in Figure 2. In the training model part, the dataset is first pre-processed to obtain sentence pairs, then a vector representation of sentences and sentence pairs is generated using the pre-trained language model, and finally a sentence coherence evaluation model integrating repetitive words features is developed after training. In the coherence evaluation part, a pair of official document sentences is fed into the trained model and the output is a coherence score.

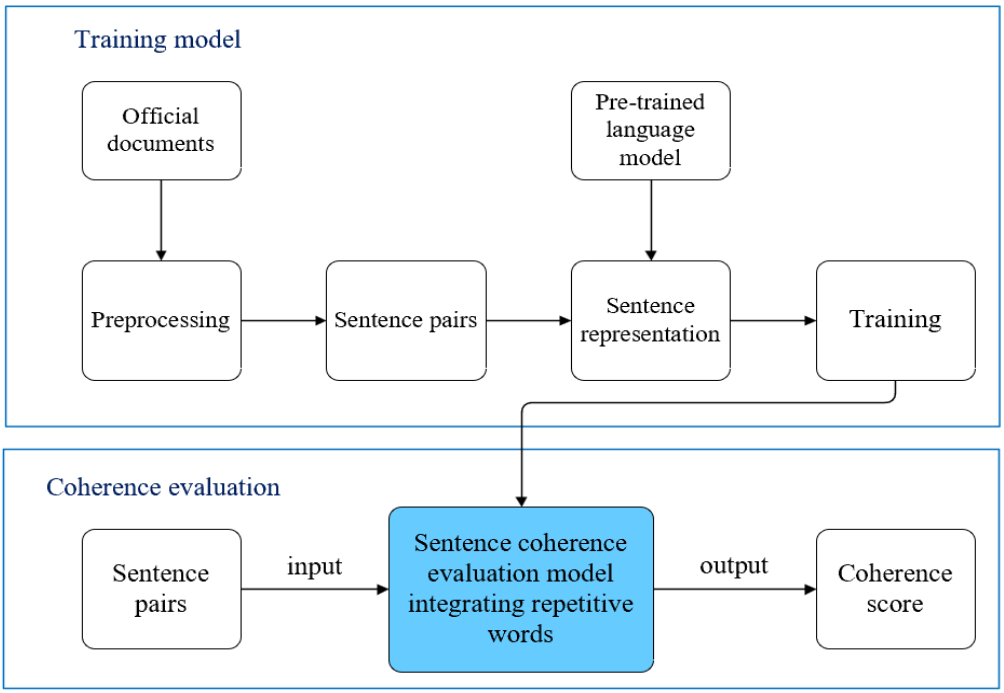

Figure 2. The research framework diagram for this paper.

3. Materials and methods

To ensure the coherence and quality of automatically generated documents, this paper conducts research on the evaluation of coherence in sentence structures of official documents. After analysis, we found that Regulations and Measures documents exhibit significant similarities in their writing styles.

1. Regulations

(1) The school has developed a fire emergency plan that defines the duties of each person in the school and their responsibilities for alerting, evacuating and fighting incipient fires, and rehearses it at least once a term. (2) The school should have a specific fire emergency plan for important events such as festivals, sports and cultural events.

2. Measures

(1) The energy-saving authorities should publish the reporting telephone number and e-mail address, after receiving energy-saving reports, should be registered truthfully. (2) For specific illegal facts, clues clear and accompanied by evidence of material reports, energy-saving authorities should be accepted in a timely manner, and the request should be provided in accordance with telephone number and e-mail address to reply.

Figure 3. Examples extracted from officially published documents.

The repetitive words we use are repeated mainly in the form of nouns. Figure 3 displays a fragment extracted from officially published documents. Example 1 consists of two consecutive sentences selected from a Regulations on school fire safety management. These sentences contain some repetitive words such as school, fire, emergency and plan, and both describe matters related to "fire

safety". Example 2 is composed of two consecutive sentences taken from the Measures on energy management, describing matters related to "energy-saving reporting". Apart from the similarity in sentence structure to Example 1, both sentences also contain many repetitive words such as energy-saving, authorities, telephone number and e-mail address. The two examples exhibit the characteristics commonly found in Regulations documents: the semantic information of adjacent sentences is related and logically coherent, and many words are repeated. Our model will fully utilize these characteristics to avoid errors caused by solely relying on textual features or semantic information to evaluate coherence.

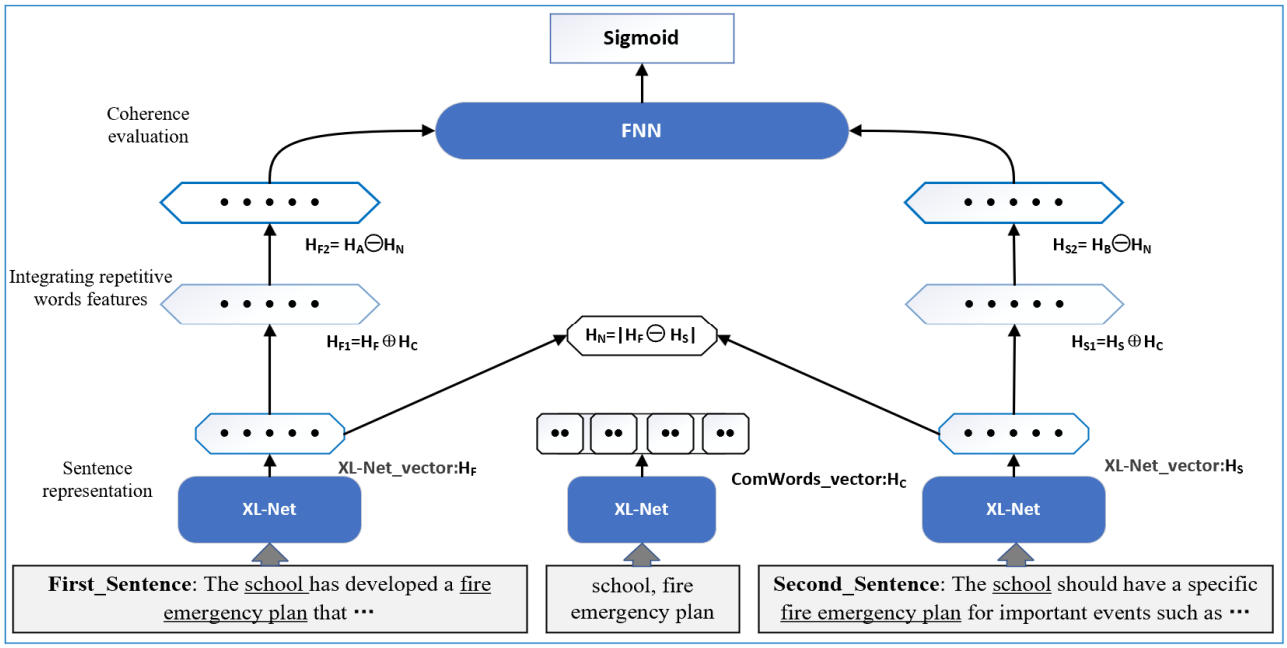

Figure 4. Coherence evaluation model integrating repetitive words features.

Figure 4 depicts the model constructed in this paper, the process is divided into the following steps.

Step 1: Extract the repetitive words in adjacent sentences, input them into the XL-Net pre-training model to obtain vector representations: XL-Net_vector (H_F, H_S) and ComWords_vector (H_C).

Step 2: The two sentence vectors are subjected to a dot-difference operation to obtain the noise vector representation of the sentences (H_N).

Step 3: Do dot-addition operation of two sentence vectors with the repetitive words feature vector respectively, to obtain sentence vectors integrating repetitive words features (H_{F1}, H_{S1}).

Step 4: The two vectors (H_{F1}, H_{S1}) are operated dot-difference with the noise vector (H_N), and the results (H_{F2}, H_{S2}) are stitched together.

Finally, after concatenating the vectors, they are fed into a Feedforward Neural Network (FNN) where they undergo calculations and mapping function transformations through the hidden layers. The output is a coherence score between 0 and 1, with higher values indicating greater coherence between sentences.

3.1. Sentence representation

The semantic coherence between sentences in official documents is not as evident and compact

as in essays or novels, which increases the difficulty of model learning. Moreover, sentences often contain words that indicate temporal sequence of events, such as "before submitting", "after receiving" and "following the regulations below". The positions of these words in sentences, as well as the order of arrangement between words, are strongly related to sentence coherence. Given these characteristics, we choose to use the XL-Net pre-trained language model to generate sentence and repetitive words vector representations.

XL-Net (Yang et al. [21]) is a bidirectional autoregressive language model (AR) proposed by a joint team from CMU and Google Brain in 2019, which is an upgraded version of BERT (Devlin et al. [22]). This model has achieved the best performance in various tasks such as machine question answering and natural language inference, and has shown significant improvement in reading comprehension tasks for long texts, thanks to its Permutation Language Modeling (PLM) training approach in the pre-training phase.

In the pre-training phase, suppose the input sequence is X and its sequence length is L. Then the sequence X has a total of $L!$ permutations. Let all permutations be O, and o is one of them, $o \in O$. Using o_i to denote the i-th word in the arrangement o, and $o<i$ to denote the first i-1 words, the maximum likelihood optimization method for PLM is as shown in Eq (1).

$$max\ E_{o\sim O}\left[\sum_{i=1}^{L} log\ p\left(x_{o_i}|x_{o_{<i}}\right)\right] \tag{1}$$

According to this optimization, PLM maximizes the likelihood probability on the training data by adjusting the model parameters, demonstrating that the autoregressive language model can also be implemented in both directions.

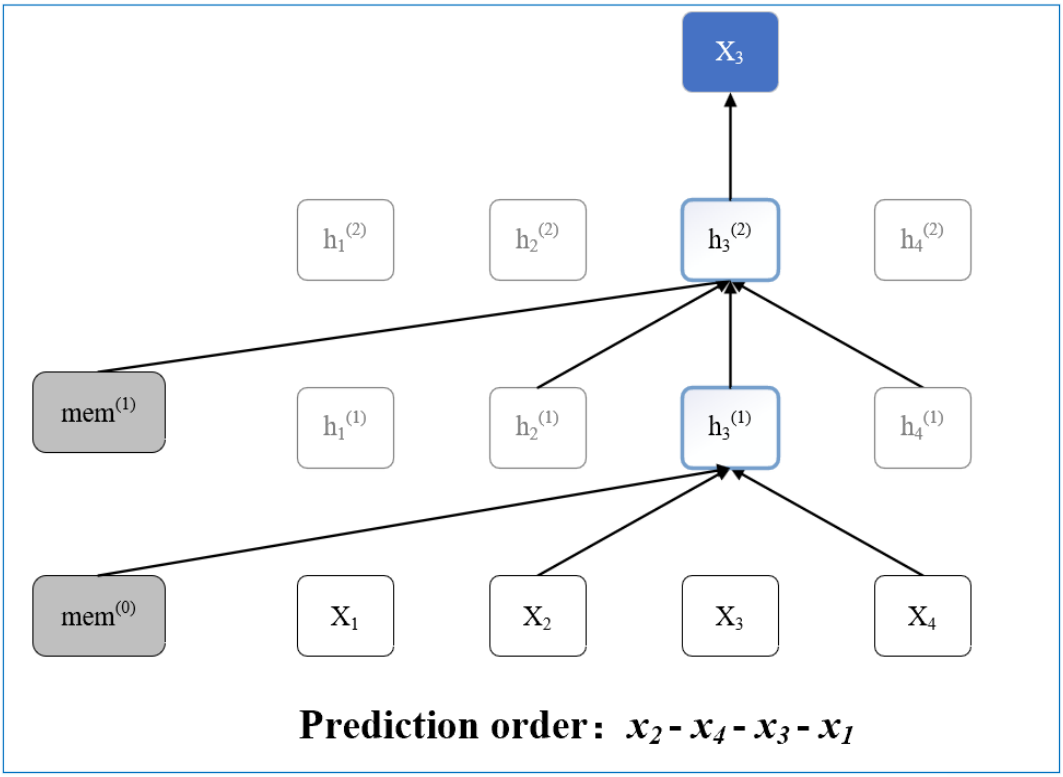

Figure 5. PLM training example.

As shown in Figure 5, suppose a sentence consists of x_1-x_2-x_3-x_4, and $h_i^{(j)}$ represents their corresponding word vectors. If the word to be predicted is x_3, x_1, x_2 and x_4 will be randomly arranged, and then some sequences are randomly selected among all possible arrangements for training (the sequence shown in Figure 5 is x_2-x_4-x_3-x_1), the word x_3 to be predicted can see both x_1 and x_2 above and x_4 below, so that the semantic information of the bidirectional context is taken into account.

The PLM training method adopted by the XL-Net model enables a comprehensive consideration of the arrangement order of words in sentences in official documents, thereby facilitating the implementation of deep bidirectional encoding of sentences, which is highly aligned with coherence evaluation tasks. Consequently, this study employs a pre-trained Chinese XL-Net model to generate vector representations of sentences, which contributes to the retention of the entire semantic and logical information of the context.

3.2. Sentence representation integrating repetitive words features

Traditional neural network-based approaches can capture deep semantic information in sentences but lack further integration of text features. In addition to the smooth semantic connection between adjacent sentences, documents such as Regulations and Measures have a common characteristic where many words are repetitive in adjacent sentences. To address this issue, we introduce the repetitive words features to overcome the shortcomings of traditional neural network methods.

Dot-difference is a subtraction between elements in two vectors by position. Previously, Mou et al. [15] and Xu et al. [17] performed dot-difference operations on feature vectors of adjacent sentences and combined the resulting calculation as a new feature to enable the model to learn the semantic logical relationships between them. But we guess that the difference in semantic features between adjacent sentences increases the difficulty of learning coherent samples for the model and is a major factor affecting the discriminative ability of the model. For example, Fire Safety, Fire Safety Responsibility and Fire Safety Accountability contain different information. Therefore, we define the difference of semantic features between adjacent sentences as noise, and the noise vector is calculated by dot-difference operations as shown in Eq (2).

$$H_N = \left| H_F \ominus H_S \right| \tag{2}$$

where H_F and H_S represent the two sentence vectors, and H_N is the noise vector of the two sentences.

Entities are extracted and used as a feature to enhance the model's ability to capture linking relationships between sentences. However, the quantity and distribution of entities in adjacent sentences in literature differ greatly, and the features are not significant. Therefore, we fully consider the special characteristics of literature sentences and believe that the number of repetitive words in adjacent sentences is relatively stable semantically, and the features are easy to capture. The repetitive words are used as a feature and integrated with the sentence vector. The calculation formula is shown in Eqs (3) and (4).

$$H_{F1} = H_F \oplus H_C \tag{3}$$

$$H_{S1} = H_S \oplus H_C \tag{4}$$

where H_C represents the repetitive words vector, the calculated results H_{F1} and H_{S1} are the sentence vectors of integrated repetitive words features, respectively.

To reduce the influence of noise on coherence evaluation, we perform noise reduction for H_{F1}

and H_{S1} with noise vector H_N calculated by Eq (2) respectively. Noise reduction are performed by dot-difference operations as shown in Eqs (5) and (6).

$$H_{F2} = H_{F1} \ominus H_N \tag{5}$$

$$H_{S2} = H_{S1} \ominus H_N \tag{6}$$

After fusing repetitive words features and noise reduction, the two sentence vectors are finally represented as H_{F2} and H_{S2}, and they are used as the input of the coherence evaluation module.

3.3. Coherence evaluation

The coherence evaluation module is a Feedforward Neural Network consisting of an input layer, a hidden layer and an output layer, whose task is to evaluate the coherence of the sentence vectors (H_{F2}, H_{S2}). The process is to splice H_{F2} and H_{S2} into a new vector E, and define the corresponding label y^E for the vector E. If E is a positive sample, the label y^E is 1, if negative, the label y^E is 0. Then the features of the input data are calculated and divided in the hidden layer so as to fit the labels corresponding to the data, and finally the coherence score is calculated by inputting it into the sigmoid activation function, as shown in Eq (7).

$$P_{(y_{E=(0|1)})} = sigmoid(U^T q_E + b) \tag{7}$$

The q^E is the output of the hidden layer, U^T represents a vector of size $1 \times H$, H represents the hidden layer length (set to 768), and b represents the bias. After the sigmoid function mapping, the output P is a floating point number belonging to the range of 0 ~ 1, a larger value represents a stronger coherence of E.

4. Experimental setup

4.1. Dataset

4.1.1. Official documents dataset

The goal of this work is to address the issue of lack of coherence between sentences in automatically generated official documents, thereby improving the quality and readability of the generated text. To achieve this, official documents of the regulation and method categories were selected as the dataset, and were classified into different domains such as economy, public transportation, and environmental protection. To ensure a balanced distribution of data, 40 official documents were selected from each domain, and were segmented into complete sentences to generate positive and negative samples. The official documents dataset contained 5066 samples, including 4559 training samples (90%) and 507 validation samples (10%). After tokenization and statistical analysis, the dataset contained about 6001 different words, and the longest sentence contained 107 characters (including punctuation).

4.1.2. THUCNews

The THUCNews dataset provided by the Natural Language Processing Group at Tsinghua

University was selected as the public dataset which covers 10 domains such as politics, lifestyle, and education. To ensure a balanced distribution of data, 60 complete news articles were selected from each domain, and were segmented into sentences to generate positive and negative samples. The news dataset contained 5070 samples, including 4563 training samples (90%) and 507 validation samples (10%). After tokenization and statistical analysis, the corpus contained about 14,138 different words, and the longest news sentence contained 126 characters (including punctuation).

4.1.3. Preprocessing and setting

In the preprocessing stage, the experimental data were divided into training and validation sets, each containing positive and negative samples. Positive samples were created by concatenating adjacent sentences in official documents to form coherent sentence pairs, and then replacing the second sentence with a random sentence from the corpus to generate negative samples that lacked coherence.

The experiment in this paper was implemented based on the PyTorch framework, and the Jieba Chinese word segmentation tool was used to tokenize the sentences. The string matching algorithm was used to extract duplicate words. The Chinese pre-trained language model used in the experiment was XLNet_base model released by the joint laboratory of Harbin Institute of Technology and iFLYTEK. The length of the word embedding was 768, the length of the FNN hidden layer was 768, the initial learning rate of the model was 1e-5, and the batch size was 20. In the non-pre-trained model approach, the length of the word embedding was 256, the length of the FNN hidden layer was 512, the initial learning rate was 1e-4, and the batch size was also 20.

4.2. Implementation details

4.2.1. Loss function

The cross-entropy function is chosen as the loss function. The cross entropy is used to determine the closeness of the actual output to the desired output, and its smaller value represents the closer probability distribution, as shown in Eq (8).

$$Loss(\delta) = \frac{1}{M}\sum -\{y_E logP(y_E = 1) + (1 - y_E)\log[1 - P(y_E = 1)]\} + \frac{Q}{2M}\sum_{\vartheta\in\delta}\vartheta^2 \tag{8}$$

where δ represents the parameters in the model, M denotes the total number of training samples, y^E is the label corresponding to the samples, and Q is a regularization term to prevent overfitting of the model.

4.2.2. Optimization

The Adam optimizer is used to optimize the loss function. The Adam optimization algorithm absorbs the advantages of Adaptive Gradient Descent and Momentum Gradient Descent, and can update the model parameters according to the oscillation of the historical gradient and the real gradient after filtering the oscillation, which can both adapt to the sparse gradient and alleviate the gradient oscillation problem, and is a commonly used optimization algorithm at present.

4.2.3. Evaluation metric

In the related research, accuracy has been widely used as the evaluation metric, which measures

the proportion of correctly predicted samples to the total number of samples in the validation set. To compare our method with theirs, we also adopted accuracy as the evaluation metric in this experiment to assess the performance of our model by comparing the accuracy scores. For example, suppose the validation set contains i samples, and the total number of correctly predicted positive and negative samples by the model is j. The accuracy of the model can be calculated as shown in Eq (9).

$$Accuracy = \frac{j(\text{Number of samples predicted to be correct})}{i(\text{Total sample size})} * 100\% \tag{9}$$

We also use precision and recall as evaluation metrics. Recall refers to the proportion of positive samples correctly identified by the model, and precision refers to the proportion of positive samples identified by the model that are actually positive.

4.3. Contrast models

We compare our model to the state of the arts as listed below:

1) Li recurrent model

In 2014, Li et al. [14] proposed a neural network model based on distributed sentence vector representations, which utilizes an RNN to obtain a distributed vector representation of a sentence, and then constructs a graph to score the coherence of two news datasets. This model eliminates the tedious feature engineering steps in traditional methods and automatically extracts various relationships and grammar features representing coherence in the text, thus having certain advantages in capturing text coherence.

2) Entity-driven Bi-LSTM

In 2016, Du et al. [19] proposed an entity-driven Bi-LSTM coherence model that extracts adjacent sentence entity information and represents it in a distributed manner. This information is then fused into Bi-LSTM through various simple and effective vector operations. Experiments on Chinese and English datasets for sentence ordering and coherence evaluation tasks show that the performance of this model is improved compared to existing models. LSTM has shown good performance in capturing long-distance context dependencies in sentences.

3) LCD

In 2019, Xu et al. [17] proposed a Local Coherence Discourse (LCD) model that approximates the global coherence score of a text by averaging the coherence scores between continuous pairs of sentences. The model's generalization ability was validated using cross-domain transfer methods. The LCD model has a simple structure, and even with the simplest sentence encoder (average GloVe), it outperforms other methods on closed-domain datasets and all open-domain datasets. Moreover, higher accuracy can be achieved by using more powerful encoders. We reproduce this method based on LSTM as a contrast model for comparison.

4) DiscoScore

In 2022, Zhao et al. [23] proposed a model called DiscoScore for evaluating the quality of text generation. Prior to DiscoScore, BERTScore (Zhang et al. [24]) was unable to recognize coherence and could not penalize incoherent elements in the output text. Therefore, DiscoScore models coherence from different perspectives of Centering theory based on BERT and calculates coherence and factuality scores by comparing the generated text with the original text. DiscoScore has been shown to be effective in various tasks such as machine translation, summarization, and dialogue systems.

4.4. Results

Table 1 shows that our model outperforms the other four comparison models on two datasets. This is because our model not only utilizes XL-Net outstanding ability to represent sentence semantic and logical information but also integrates repetitive words features to enhance coherence between sentences. Finally, by using denoising techniques, we reduce the impact of noise on the results, thereby improving the accuracy of the model. The Li Recurrent model uses RNN to obtain the vector representation of the sentence, while the Entity-driven Bi-LSTM and LCD use LSTM to obtain the vector representation of the sentence. However, when dealing with long texts, RNN and LSTM have limited effectiveness in solving the problem of long-distance dependencies, and their feature extraction ability is inferior to that of BERT and XL-Net, resulting in slower convergence during model training. The DiscoScore models coherence from different perspectives driven by Centering theory. But if there is little coherent phenomena between sentences (such as a few number of repetitive entities), the model performs poorly.

Table 1. Comparison with the experimental results of contrast models.

Model	THUCNews			Official Documents Dataset		
	accuracy	precision	recall	accuracy	precision	recall
Li Recurrent Model	0.578	0.620	0.689	0.646	0.675	0.701
Entity-driven Bi-LSTM	0.623	0.667	0.732	0.807	0.711	0.753
LCD	0.796	0.752	0.767	0.864	0.878	0.897
DiscoScore	0.909	0.889	0.910	0.937	0.915	0.923
Ours	0.955	0.931	0.924	0.962	0.938	0.933

To avoid over-fitting leading to poor performance of the models, we conducted 10-fold cross validation experiments on both datasets and the results are shown in Table 2. We divide all samples into ten, and then use each one as the validation set and the others as the training set for training and validation.

Table 2. 10-fold cross validation experiments.

Model	THUCNews			Official Documents Dataset		
	accuracy	precision	recall	accuracy	precision	recall
Li Recurrent Model	0.563	0.622	0.678	0.629	0.677	0.703
Entity-driven Bi-LSTM	0.598	0.645	0.720	0.801	0.704	0.725
LCD	0.774	0.739	0.756	0.836	0.856	0.876
DiscoScore	0.895	0.883	0.919	0.925	0.892	0.900
Ours	0.934	0.920	0.912	0.947	0.928	0.916

It can be seen that all models perform better on the official documents dataset compared to THUCNews dataset. One possible reason is that the average sentence length in the official documents dataset is shorter, making it relatively easier for different models to overcome the problem of long-distance dependencies on the official documents dataset. Another reason is that the sentence structure in official documents is relatively uniform, while news sentence patterns are generally more diverse and random, with fewer repetitive words between adjacent sentences than in official documents.

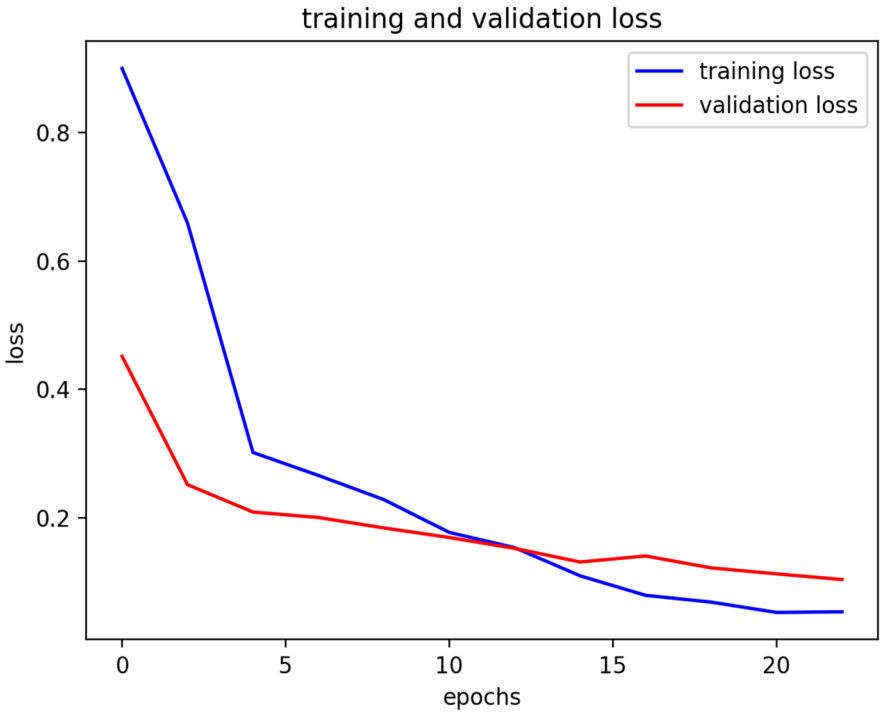

Figure 6. Training and validation loss curves.

To monitor the loss function, we use a graph to plot the training and validation loss curves where the y-axis is the loss value and the x-axis is the number of training epochs, as shown in Figure 6.

To further validate the effectiveness of our method, we compared it with XL-Net, XL-Net+entities and XL-Net+repetitive words features (named XL-Net+rwf) as baseline approaches. In addition, we implemented our method based on RNN and Bi-LSTM and compared their performances to demonstrate the effectiveness of ours. The experimental results are shown in Table 3.

Table 3. Comparison with the experimental results of baseline models.

Model	THUCNews			Official Documents Dataset		
	accuracy	precision	recall	accuracy	precision	recall
RNN	0.578	0.620	0.689	0.646	0.675	0.701
RNN+rwf+noise reduction	0.597	0.635	0.696	0.665	0.694	0.736
Bi-LSTM	0.623	0.658	0.729	0.807	0.706	0.751
Bi-LSTM+rwf+noise reduction	0.672	0.676	0.737	0.843	0.732	0.774
XL-Net	0.930	0.913	0.916	0.945	0.922	0.931
XL-Net+entities	0.925	0.894	0.899	0.941	0.911	0.919
XL-Net+rwf	0.934	0.917	0.921	0.949	0.926	0.931
XL-Net+rwf+noise reduction (Ours)	0.955	0.931	0.924	0.962	0.938	0.933

From Table 3, it can be seen that integrating repetitive words features enhances the neural network's ability to learn sentence coherence compared to methods that do not use text features or only use entities features. On the one hand, the two datasets contain a large number of entities with complex distributions that the model cannot adequately learn their dependencies. On the other hand,

adjacent sentences contain non-identical entities, which increases the noise in the model. In contrast, repetitive words have relatively stable semantics in adjacent sentences, avoiding the aforementioned problems.

In addition, the semantic information of the two sentences is more similar in positive samples, and the noise that affects coherence is smaller. Therefore, using dot-difference operation on top of integrating repetitive words features can effectively alleviate the noise impact on coherence evaluation. Due to the relatively fixed grammatical structure of official documents, our method achieved the highest accuracy on the official documents dataset, while modeling coherence for news is more challenging.

5. Conclusions

Sentence coherence is an important factor in ensuring the readability of a text. In order to improve the quality of automatically generated official documents texts, this paper focuses on the evaluation of sentence coherence in official documents. After analyzing the characteristics of Regulations and Measures documents, we propose a coherence evaluation model integrating repetitive words features, which introduces repetitive words features in adjacent sentences into the research of coherence evaluation for the first time. Experimental results show that our model outperforms the compared models on two datasets. However, the elements of sentence coherence are multifaceted, and the features considered in this paper are relatively singular. Therefore, it is worth exploring multiple coherence features to further improve the accuracy of the model.

Conflict of interest

The authors declare there is no conflict of interest.

References

1. S. Prabhu, K. Akhila, S. Sanriya, A hybrid approach towards automated essay evaluation based on BERT and feature engineering, in *2022 IEEE 7th International Conference for Convergence in Technology (I2CT)*, IEEE, Vadodara, India, (2022), 1–4. https://doi.org/10.1109/I2CT54291.2022.9824999
2. S. Jeon, M. Strube, Centering-based neural coherence modeling with hierarchical discourse segments, in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, ACL, Online, (2020), 7458–7472. https://doi.org/10.18653/v1/2020.emnlp-main.604
3. X. Tan, L. Zhang, D. Xiong, G. Zhou, Hierarchical modeling of global context for document-level neural machine translation, in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, ACL, Hong Kong, China, (2019), 1576–1585. https://doi.org/10.18653/v1/D19-1168
4. Q. Zhou, N. Yang, F. Wei, S. Huang, M. Zhou, T. Zhao, Neural document summarization by jointly learning to score and select sentences, in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, ACL, Melbourne, Australia, (2018), 654–663. https://doi.org/10.18653/v1/p18-1061

5. Y. Diao, H. Lin, L. Yang, X. Fan, Y. Chu, D. Wu, et al., CRHASum: extractive text summarization with contextualized-representation hierarchical-attention summarization network, *Neural Comput. Appl.*, **32** (2020), 11491–11503. https://doi.org/10.1007/s00521-019-04638-3
6. P. Yang, L. Li, F. Luo, T. Liu, X. Sun, Enhancing topic-to-essay generation with external commonsense knowledge, in *Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL)*, ACL, Florence, Italy, (2019), 2002–2012. https://doi.org/10.18653/v1/p19-1193
7. X. L. Li, J. Thickstun, I. Gulrajani, P. Liang, T. B. Hashimoto, Diffusion-LM improves controllable text generation, *arXiv preprint*, (2022), arXiv:2205.14217. https://doi.org/10.48550/arXiv.2205.14217
8. D. Parveen, H. M. Ramsl, M. Strube, Topical coherence for graph-based extractive summarization, in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, ACL, Lisbon, Portugal, (2015), 1949–1954. https://doi.org/10.18653/v1/d15-1226
9. L. Logeswaran, H. Lee, D. R. Radev, Sentence ordering and coherence modeling using recurrent neural networks, in *Proceedings of the 8th AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI, Palo Alto, USA, **32** (2018), 5285–5292. https://doi.org/10.1609/aaai.v32i1.11997
10. Y. Liu, M. Lapata, Learning structured text representations, *Trans. Assoc. Comput. Ling.*, **6** (2018), 63–75. https://doi.org/10.1162/tacl_a_00005
11. R. Barzilay, M. Lapata, Modeling local coherence: an entity-based approach, in *43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, ACL, Michigan, USA, (2005), 141–148. https://doi.org/10.3115/1219840.1219858
12. A. Louis, A. Nenkova, A coherence model based on syntactic patterns, in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (*EMNLP-CoNLL)*, ACL, Jeju Island, Korea, (2012), 1157–1168.
13. D. Parveen, M. Mesgar, M. Strube, Generating coherent summaries of scientific articles using coherence patterns, in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (*EMNLP)*, ACL, Austin, USA, (2016), 772–783. https://doi.org/10.18653/v1/d16-1074
14. J. Li, E. H. Hovy, A model of coherence based on distributed sentence representation, in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, ACL, Doha, Qatar, (2014), 2039–2048. https://doi.org/10.3115/v1/d14-1218
15. L. Mou, R. Men, G. Li, Y. Xu, L. Zhang, R. Yan, et al., Recognizing entailment and contradiction by tree-based convolution, *arXiv preprint*, (2016), arXiv:1512.08422. https://doi.org/10.48550/arXiv.1512.08422
16. K. Luan, X. Du, C. Sun, B. Liu, X. Wang, Sentence ordering based on attention mechanism, *J. Chin. Inf. Technol.*, **32** (2018), 123–130. https://doi.org/10.3969/j.issn.1003-0077.2018.01.016
17. P. Xu, H. Saghir, J. S. Kang, T. Long, A. J. Bose, Y. Cao, et al., A cross-domain transferable neural coherence model, in *Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL)*, ACL, Florence, Italy, (2019), 678–687. https://doi.org/10.18653/v1/p19-1067
18. F. Xu, S. Du, M. Li, M. Wang, An entity-driven recursive neural network model for Chinese discourse coherence modeling, *Int. J. Artif. Intell. Appl.*, **8** (2017), 1–9. https://doi.org/10.5121/ijaia.2017.8201

19. S. Du, F. Xu, M. Wang, An entity-driven bidirectional LSTM model for discourse coherence in Chinese, *J. Chin. Inf. Technol.*, **31** (2017), 67–74. https://doi.org/10.3969/j.issn.1003-0077.2017.06.010
20. K. Liu, H. Wang, Research on automatic summarization coherence based on discourse rhetoric structure in Chinese, *J. Chin. Inf. Technol.*, **33** (2019), 77–84. https://doi.org/10.3969/j.issn.1003-0077.2019.01.009
21. Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, Q. V. Le, XLNet: generalized autoregressive pretraining for language understanding, in *Annual Conference on Neural Information Processing Systems* (*NeurIPS)*, (2019), 5754–5764.
22. J. Devlin, M. W. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (*NAACL-HLT)*, ACL, Minneapolis, USA, (2019), 4171–4186. https://doi.org/10.18653/v1/n19-1423
23. W. Zhao, M. Strube, S. Eger, Discoscore: evaluating text generation with bert and discourse coherence, *arXiv preprint*, (2022), arXiv:2201.11176. https://doi.org/10.48550/arXiv.2201.11176
24. T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, Y. Artzi, BERTScore: evaluating text generation with BERT, *arXiv preprint*, (2019), arXiv:1904.09675. https://doi.org/10.48550/arXiv.1904.09675

AIMS Press