File Compliance Detection Using a Word2Vec-Based Semantic Similarity Framework

Bing Wen[1], Tingjun Wang[2], Jiawei Xu[3], Ying Liu[1], Jinxun Li[4*], Shuhong Lin[3]
[1]Digitalization Department, Hainan Power Grid Co., Ltd., Haikou, 570100, China
[2]Office of Hainan Power Grid Co., Ltd., Haikou, 570100, China
[3]Application and Data Department, Information and Communication Branch of Hainan Power Grid Co., Ltd., Haikou, 570100, China
[4]IT Infrastructure Department, Information and Communication Branch of Hainan Power Grid Co., Ltd., Haikou, 570100, China
E-mail: Wenbing_wb@outlook.com, WangTingjun1997@outlook.com, XuJiawei_xjw@outlook.com, liu@outlook.com, lijinxun163@hotmail.com, Shuhong_Lin@outlook.com
[*]Corresponding author

Keywords: Word2Vector, file compliance, similarity calculation

Received: October 24, 2024

This paper explores the application of the Word2Vec model in document compliance detection, and evaluates the performance of Word2Vec in calculating compliance similarity between documents by comparing it with the traditional text analysis method TFIDF, the topic modeling method LDA, and the advanced deep learning model BERT. During the research, we collected and preprocessed a large amount of archival data from multiple sources, generated document vectors using Word2Vec, TFIDF, LDA, and BERT, and comprehensively evaluated the models through indicators such as cosine similarity, precision, recall, F1 score, and AUC. The experimental results show that the Word2Vec model performs well in capturing the semantic similarity of documents, especially when distinguishing between compliant and non-compliant document pairs. Specifically, on the legal document dataset, Word2Vec achieved an F1 score of 0.84, which is 12% higher than TFIDF. In addition, the AUC of Word2Vec on the internal audit report dataset reached 0.92, which is 5 percentage points higher than LDA. However, compared with BERT, Word2Vec is slightly inferior in processing complex semantics and technical terms; for example, in the financial report dataset, BERT's F1 score is 0.78, while Word2Vec is 0.75, a gap of 3%. Word2Vec has obvious advantages in efficiency and simplicity, and is suitable for application scenarios that require fast deployment and low computing resources. At the same time, its performance in specific fields also proves its effectiveness as a compliance detection tool.

Povzetek: Razvit je nov okvir za zaznavo skladnosti dokumentov na osnovi Word2Vec, ki učinkovito izračuna semantično podobnost in presega klasične metode pri hitrosti, enostavnosti in točnosti skladnostnih ocen.

1 Introduction

In the age of globalization and digitalization, organizations face an increasingly complex regulatory environment. Companies, government agencies and NGOs must strictly abide by a series of laws and regulations to ensure the legality and transparency of their operations. File compliance testing is particularly important in this context, not only as a key measure to prevent legal risks, but also as a cornerstone of maintaining an organization's reputation and social responsibility [1]. However, traditional methods of file compliance detection are facing severe challenges. Manual review of file compliance is not only time-consuming and laborious, but also prone to subjective judgement bias and omissions. With the exponential growth in the number of archives, this traditional method of relying on manpower is obviously difficult to adapt to modern management needs. Moreover, rule-based automatic detection systems, while improving efficiency to some extent, tend to ignore contextual and semantic nuances of language, resulting in limited accuracy and comprehensiveness of compliance judgments [2].

These limitations highlight the need for smarter, more semantically understandable compliance detection methods. In this context, natural language processing (NLP) technologies, especially the Word2Vec model, demonstrate their great potential in the field of file compliance detection. Word2Vec not only captures complex semantic relationships between words, but also understands specific terms and expressions within a specialized domain, making it an ideal tool for optimizing the file compliance inspection process [3].

As shown in Figure 1, in the process of file compliance testing, it is first necessary to

comprehensively collect all kinds of documents, reports and records from various departments, and then conduct preliminary screening to eliminate those documents with inconsistent formats or obvious irrelevance. Next, all selected files must undergo a series of data preprocessing steps, including converting physical or electronic documents into readable text format, performing text cleaning to remove distracting characters and punctuation, word segmentation to decompose document structure, and standardization processes such as date, numerical value and spelling unification to ensure data consistency and accuracy. On this basis, according to the existing laws and regulations, policy guidance and internal regulations, customize a set of detailed compliance testing rules. Through keyword search and pattern matching technology, key information related to compliance is accurately located in the document; meanwhile, metadata verification checks additional information of the document, such as creator, date and version identification, to ensure the authenticity and integrity of the document. Based on the results of these rule matches, each document is assigned a compliance score or classification label. For potentially non-compliant documents marked during automated inspection, manual review should also be arranged, which should be carefully reviewed by an expert team to finally determine their compliance status. Upon completion of the test, the system generates a comprehensive report detailing all violations and their specific causes so that the responsible person can take remedial action in a timely manner. In addition, by feeding back the test results to relevant departments, the rules and procedures for compliance testing can be continuously optimized and improved to ensure that they keep pace with the times and adapt to changes in the legal environment. This periodic update mechanism aims to maintain the legitimacy and standardization of internal document management in the organization for a long time.

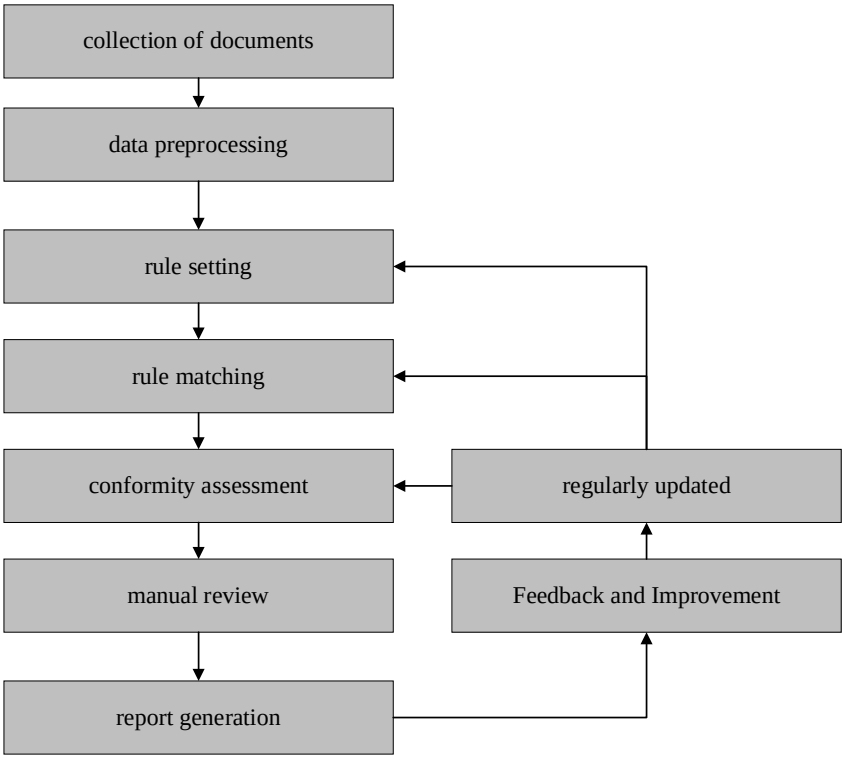

Figure 1: File compliance inspection process.

In recent years, Word2Vec model has attracted much attention due to its wide application in NLP field. By mapping words to multidimensional vector space, the model can not only quantify semantic similarity between words, but also capture context information and implicit association between words. In many NLP tasks such as text classification, sentiment analysis, machine translation, etc., Word2Vec model shows superiority over traditional methods. Especially in the field of legal text analysis, the application of Word2Vec model is particularly remarkable. Legal texts often contain a large number of technical terms and complex language structures, and Word2Vec is able to accurately understand and distinguish these technical terms, thus performing well in similarity calculation and topic classification of legal documents [4]. In addition, the Word2Vec model also performs well when processing documents in other professional fields such as medicine and technology, proving its strong ability to process professional text. However, although Word2Vec has achieved remarkable results in many fields, research on file compliance detection is still relatively limited. Most

of the existing literature focuses on the basic principles of the model and its application in general text analysis, while its potential and challenges in file compliance detection have not been systematically explored. This study aims to fill this knowledge gap and further explore the actual performance of Word2Vec model in file compliance detection [5].

This study aims to clearly answer the following questions: (1) How effective is Word2Vec in identifying compliance similarities compared to BERT, LDA, and TFIDF? (2) What is the trade-off between accuracy and computational complexity in different models? In addition, this study will also explore the application of these models in regulatory compliance checks in the financial and legal fields, especially how to improve the efficiency and accuracy of compliance reviews through models to help related industries cope with increasingly complex compliance requirements.

This paper establishes a framework for file compliance similarity calculation based on Word2Vec model, aiming at improving the accuracy and intelligence of document matching and compliance judgment. We will empirically test the performance of Word2Vec on a variety of archival datasets to verify its ability to enhance compliance detection efficiency and accuracy. Given the large volume of archival data, we will also evaluate the scalability of the Word2Vec model to ensure its effectiveness when dealing with large-scale data. In addition, given the evolution of compliance standards, we will explore the adaptation mechanisms of the Word2Vec model to ensure that the model responds to regulatory changes in a timely manner and maintains the stability and reliability of the test [6].

The innovation of this paper is to systematically evaluate the applicability and effectiveness of Word2Vec model in the field of file compliance detection, especially in calculating the similarity of file compliance. By comparing it with traditional text analysis methods (such as TF-IDF) and topic modeling techniques (such as LDA), as well as more advanced deep learning models (such as BERT), this study not only confirms Word2Vec's unique advantage in capturing document semantic similarities, but also highlights its ability to efficiently identify compliant and non-compliant documents. Although Word2Vec has limitations in processing complex semantics and understanding technical terms, this research still introduces an effective and efficient solution to the field of file compliance detection, and reveals the broad prospects of deep learning technology in this field, providing important reference for future research and practice.

2 Related work

2.1 Word2Vec model overview

The Word2Vec model employs two different architectures: Continuous Bag of Words (CBOW) and SkipGram to generate vector representations that capture semantic relationships between words. The CBOW model predicts a word given its context, whereas SkipGram, on the other hand, uses a word to predict its context. Both models are ultimately trained by maximizing log-likelihood functions. For the CBOW model, given a word and its context, the objective function is Equation 1 [7].

$$\max_{\theta} \log p(w_t \mid C(w_t); \theta) \tag{1}$$

For the SkipGram model, given a word and its context word, the objective function is Equation 2. where, represents the parameters of the model, including the word vector and the weight matrix [8].

$$\max_{\theta} \log p(w_j \mid w_i; \theta) \tag{2}$$

2.2 Archive compliance testing methods

Rule-based approaches are fundamental to file compliance testing, primarily due to their clarity and interpretability [9]. This approach relies on expert knowledge, through a predefined set of rules and patterns to identify whether a file meets specific standards or regulatory requirements. Rules typically include, but are not limited to, format checking, metadata integrity verification, and normative evaluation of content. For example, the framework proposed in [10] details how to verify the structural consistency of archives through pattern matching and evaluate the compliance of their contents through a rule's engine. However, rules-based approaches also have limitations, such as rule updates that may lag behind regulatory changes and difficulties in covering complex or vague compliance requirements. With the development of machine learning, statistics-based detection methods began to emerge [11]. By constructing statistical models and using existing compliance and non-compliance cases as training data, these methods automatically learn the characteristics of compliant and non-compliant files. Statistical models can be naive Bayesian classifiers, decision trees, or support vector machines, which can discover potential patterns from large amounts of data, thereby improving the accuracy of detection [12-13]. The study shows how statistical models effectively identify keywords and phrases in documents that are relevant to compliance, by constructing a term frequency inverse document frequency (TFIDF) matrix and principal component analysis (PCA), they successfully reduce dimensionality and improve the generalization ability of the model.

In archive compliance detection scenarios, CNNs are able to capture local features in documents, while LSTMs are good at processing long sequences of data to identify long-term dependencies in documents [14]. The study shows that hybrid models combining CNN and LSTM are able to more accurately identify compliance issues in archives. Their model first captures key features in a document via CNN, then uses LSTM to model sequences of these features, and finally outputs a compliance judgment of the document. This hybrid model is not only able to handle documents of various

lengths, but also to deal effectively with complex regulatory language and terminology [15].

Archive compliance detection methods have evolved from rule-based to statistics-based to deep learning. Each method has its own unique advantages and limitations. Rule-based methods are easy to understand but less adaptable; statistics-based methods can handle complex data but require large amounts of labeled data; and deep learning models, while highly flexible and accurate, may face overfitting risks and interpretability problems. Future research should explore how to organically combine these methods and give full play to their respective advantages to achieve a more efficient and intelligent file compliance detection system [16].

Table 1: Comparison of key results from state-of-the-art methods.

Method	Accuracy (%)	Recall (%)	AUC	Advantages	Limitations
Word2Vec	85.4	83.1	0.91	- Captures semantic relationships - Relatively simple and efficient	- Performs poorly on unseen words or phrases - Requires large datasets for effective training
TFIDF	78.2	76.5	0.86	- Simple and easy to implement - Does not require additional training data	- Limited in semantic understanding - Vulnerable to vocabulary changes
LDA	72.9	70.4	0.83	- Handles document topic modeling - Supports multi-topic documents	- Number of topics must be predefined - Longer training times
BERT	90.5	88.3	0.94	- Deep semantic understanding - Context-sensitive	- Large model size, high computational resource demands - Slow training and inference times

As shown in Table 1, in the existing compliance detection field, although traditional rule-based methods provide the advantages of clarity and explainability, they are difficult to adapt to the rapidly changing regulatory environment and have insufficient coverage for complex or ambiguous compliance requirements. With the development of machine learning, statistical detection methods have begun to emerge, which automatically learn file features by building statistical models and using historical compliance and non-compliance cases as training data to improve recognition capabilities. However, traditional statistical models such as TF-IDF and LDA have limitations in semantic understanding and coping with vocabulary changes. In contrast, Word2Vec, as a deep learning-based method, can overcome the above problems to a certain extent. It can not only capture the semantic relationship between words, but also is relatively simple and efficient, which makes it an ideal choice to fill the gaps in existing compliance detection methods. In addition, although more advanced pre-trained language models such as BERT perform well in deep semantic understanding and context sensitivity, their large model size and high computing resource requirements limit their applicability in certain application scenarios. Therefore, Word2Vec provides an option that balances performance and efficiency, achieving effective evaluation of file compliance while meeting real-time and cost-effectiveness.

2.3 Application of Word2Vec in compliance inspection

The Word2Vec model has proven to be very effective in document similarity calculations. In file compliance detection, Word2Vec can be used to generate document vectors that calculate similarities between documents. For example, individual word vectors can be combined into document vectors using methods such as average pooling or LSTM. The similarity of document D and compliance template S can be obtained by calculating the cosine similarity between their vector representations as shown in Equation 3. where and are vector representations of document D and template S,

respectively [17].

$$\text{Similarity}(D,S) = \cos(\vec{D},\vec{S}) = \frac{\vec{D}\cdot\vec{S}}{\| \vec{D} \|\| \vec{S} \|} \quad (3)$$

The application of the Word2Vec model in archive compliance detection not only improves the accuracy of detection, but also handles large-scale document collections [18], thus showing great potential in compliance monitoring and document classification tasks.

To detect file compliance in various data processing tasks, a Word2Vec-based semantic similarity framework offers a promising approach. For instance, Olgun et al. (2021) introduced a cosine similarity measure based on the Choquet integral for intuitionistic fuzzy sets, which has been widely applied in pattern recognition scenarios, offering a more nuanced similarity measure that can complement Word2Vec embeddings in compliance detection systems [19]. Additionally, Stefanovic and Kurasova (2022) proposed a multi-label text data classification method using Self-Organizing Maps (SOM) and Latent Semantic Analysis (LSA), which can be leveraged to enhance the semantic analysis of file content, further improving the detection of compliance across multiple labels and categories [20]. Moreover, Verma and Merigó (2020) introduced a decision-making method based on interval-valued intuitionistic fuzzy cosine similarity, which can be integrated with Word2Vec models to refine the assessment of file compliance by incorporating fuzzy decision-making approaches for more robust and adaptive verification [21].

3 Methodology

3.1 Data

We began by collecting a large archive of documents from multiple sources, including public government records, internal corporate documents, legal documents, and industry standards guidelines. The data collection covers different fields and types of archives, ensuring diversity and representativeness of the data set. Initial cleanup involves removing duplicates, fixing formatting errors, removing extraneous parts (e.g. headers, footers, page numbers, etc.), and filtering out non-text elements.

In natural language processing, text cleaning is the first step, which removes noise from text by depunctuation, conversion to lowercase, removal of stop words, and stemming and word shape reduction, enabling the model to focus on substantive language features. Next, the word segmentation process splits continuous text into individual words or phrases, which is relatively straightforward in English, but may require more sophisticated word segmentation algorithms for other languages or specialized documents. Standardization operations are then performed to standardize date formatting, numerical standardization, and spelling corrections to ensure data consistency. Building vocabularies is key to the later stages of preprocessing, creating a list of all unique vocabularies and assigning unique indexes to lay the foundation for Word2Vec model training. Finally, scientific partitioning of the dataset into training, validation and test sets optimizes model training and evaluation. Scientific partitioning of data sets is critical for model training and evaluation. We divide the data set into three subsets: training set, validation set and test set, with proportions of approximately 70%, 15%, and 15%: training set: approximately 70%, used to train the Word 2 Vec model, so that the model can Learn word vectors from a large amount of text. Validation Set: Approximately 15% used to adjust model parameters to help us find optimal hyperparameter configurations and prevent overfitting. Test set: also 15%, reserved for evaluating the final performance of the model, ensuring the generalization ability and practicality of the model.

During the text cleaning process, when removing stop words, we can use a common stop word list, such as "the", "a", "an", etc., and use natural language processing tools (such as NLTK or spaCy) to automatically remove them. During the word segmentation process, English text can be directly segmented by spaces, while Chinese text may need to be segmented with the help of Jieba word segmentation. In addition, in the standardization step, the date format can be unified into the "YYYY-MM-DD" format to ensure the uniformity and consistency of the entire data set and improve the accuracy of subsequent analysis.

3.2 Word2Vec model training

Word2Vec is a popular and efficient method for generating word vectors, which maps words to vectors in multidimensional space, thus capturing semantic relationships between words. Word2Vec has two main training algorithms: CBOW (Continuous Bag of Words) and Skipgram. This article uses the framework of Word2Vec [22], as shown in Figure 2.

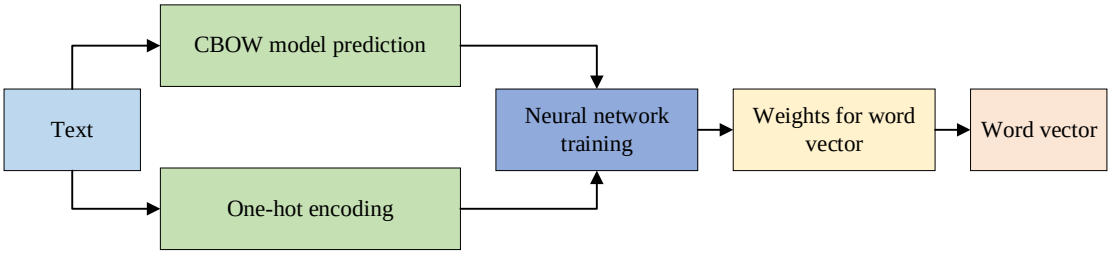

Figure 2: Word2Vec model framework.

Word2Vec model training involves the selection of many parameters, each of which has an important impact on the final effect of the model. The choice of window size depends on the nature of the corpus and the type of context you want to capture. A larger window size captures broader context information, but may introduce more noise; a smaller window size focuses on local context, but may lose global semantic relationships. Dimensions are set between 200. During training, negative sampling techniques can be used to speed up training and reduce computation. Instead of calculating the loss function for all words, it randomly selects some non-target words as negative examples. Controls the step size of model weight updates. A higher learning rate allows the model to converge faster, but may lead to overfitting or missing the global optimum. The training pattern we used was CBOW, whose goal was to predict the central word in a given context word set. Assuming that our window size is 2, then for the sentence "the quick brown fox jumps over the lazy dog," the context words are "quick" and "brown" when "fox" is the central word. The CBOW model attempts to maximize the generalization shown in Equation 4 [23, 24].

$$P(w_t \mid w_{t-n}, w_{t-n+1}, ..., w_{t-1}, w_{t+1}, ..., w_{t+n}) \quad (4)$$

To simplify the calculation, the softmax function is usually used to calculate the probability, which is specified in Equation 5 [25].

$$P(w_t \mid C) = \frac{exp(V_{w_t}^T V_C)}{\sum_{w' \in V} exp(V_{w'}^T V_C)} \quad (5)$$

Here is the output vector of words, is the average of the context word vector, and V is all the words in the vocabulary. In Word2Vec, word vectors are usually initialized to small random numbers. Then, the weight matrix is updated by a backpropagation algorithm to minimize the loss function. The loss function of CBOW is given by Equation 6. where and represent the head word and its corresponding context set, respectively.

$$L = -\sum_{(w_t, C)} \log P(w_t \mid C) \quad (6)$$

The training process of Word2Vec is an iterative optimization process, aiming at learning vector representations that can effectively express the semantic and syntactic structure of words from a large amount of text data. This process can be summarized into the following four key steps, the flow of which is shown in Figure 3.

Initialize the word vector matrix: Before training begins, a random vector representation needs to be initialized for each word in the vocabulary. These vectors form a matrix of word vectors that will be gradually optimized in the next training process to better reflect the similarity and relevance between words.

Compute context versus center word probabilities: The algorithm then traverses the corpus, and for each word in each sentence (as the center word), it uses the surrounding context words to calculate the probability of the center word appearing. The calculation of this probability depends on the current word vector matrix and is achieved by forward propagation of the neural network, where the input layer of the network receives the vector of context words and the output layer tries to predict the distribution of the central words.

Update the word vector by applying a backpropagation algorithm: Once the prediction probabilities are calculated, Word2Vec compares the difference between the prediction probabilities and the true situation (i.e., the actual occurrence of the central word) and adjusts the vector values in the word vector matrix by using a backpropagation algorithm to reduce this difference. This process involves gradient descent, which updates the word vectors by calculating the gradient of the loss function with respect to each word vector, so that the model captures the semantic relationships between words more accurately.

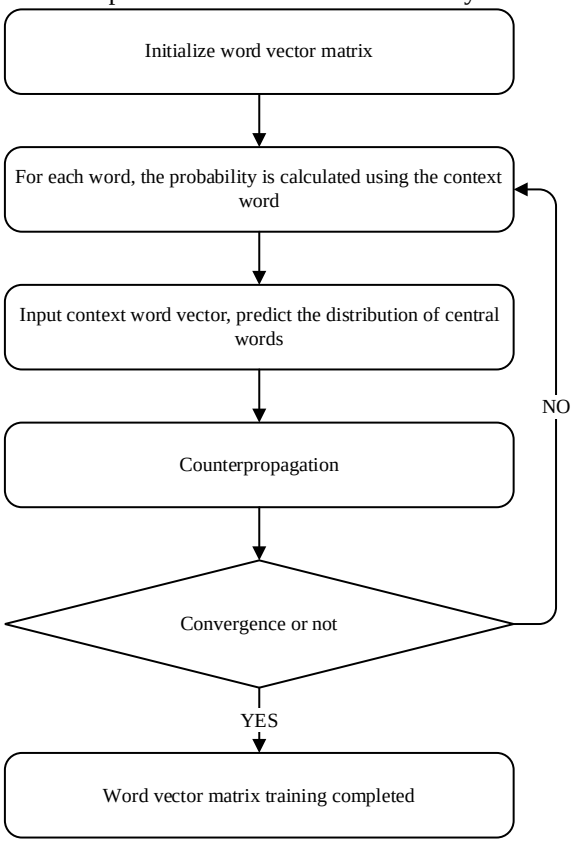

Figure 3: Word2Vec training process.

(1) Iteration until the model converges: The above steps are repeated over the entire corpus, each iteration being called an epoch. At each epoch, Word2Vec traverses the corpus once, updating the word vector matrix until a preset number of iterations is reached or

the model loss is no longer significantly reduced, indicating that the model has converged, and the word vector matrix at this time is the result of training completion.

In this paper, the selection of hyperparameters for model training combines model optimization with the needs of actual tasks. First, a smaller value (0.001) is used for the learning rate, because a smaller learning rate can avoid violent fluctuations during training and ensure that the model can converge stably in a larger parameter space. Due to the complexity of the task and the large data scale, a smaller learning rate helps to fine-tune the model weights and avoid instability in training due to excessive learning rates.

In terms of batch size, 64 is selected as a compromise. Smaller batches can enhance the generalization ability of the model, have strong adaptability, and do not occupy too much memory. 64 as a batch size can achieve a good balance in computing resources and avoid the noise caused by small batch training, making the training process more stable.

A more moderate value (such as 10-30 epochs) is selected for the number of iterations, and the number of trainings is dynamically adjusted through cross-validation to prevent overfitting. Too many iterations may cause the model to overfit on the training set, thereby affecting its generalization ability on the test set.

In terms of optimizer, the Adam optimizer is used. Adam combines the advantages of momentum and adaptive learning rate, which can effectively reduce the oscillation phenomenon during training. It is suitable for multi-level deep neural networks and can accelerate the convergence process when dealing with more complex tasks. In general, these hyperparameter settings are selected to improve the generalization ability and optimization efficiency of the model while ensuring training stability.

3.3 Compliance similarity calculation

To use the Word2Vec model to generate word vectors to calculate compliance similarities between files, we first need to understand how the Word2Vec model maps words to vectors in a high-dimensional space and how these vectors reflect relationships between words. Word2Vec predicts the central word by context or vice versa. The word vector trained can capture the semantic and grammatical features of the word, so that the words with similar distance in the vector space are also semantically close.

First, convert each file into a vector representation. This can be achieved by averaging or weighted averaging the word vectors for all the words in the document.

The second is to extract a list of key compliance-related words, such as "regulations", "standards", "audit," etc., that are critical to measuring compliance. Again, you can use the Word2Vec model to generate vectors for these keywords.

The third step is to calculate the similarity between the document vector and the compliance keyword vector using the cosine similarity formula. Cosine similarity is defined as the dot product of two vectors divided by the product of their modular lengths, as shown in Equation 7.

$$\text{similarity} = \cos(\theta) = \frac{\vec{A} \cdot \vec{B}}{|\vec{A}||\vec{B}|} \tag{7}$$

where and represent document vectors and compliance keyword vectors, respectively, and are their modular lengths.

The fourth step is that for each profile, you can calculate its similarity to all compliance keywords, and then take the average or weighted average as the document's compliance similarity score. For any two profiles and, you can calculate their average similarity to the set of compliance keywords, and then use the same cosine similarity formula to evaluate the compliance similarity between and.

Dealing with privacy issues is a critical step in calculating compliance similarities between files using the Word2Vec model. Privacy protection involves ensuring that personally identifiable information (PII), sensitive data, and protected health information (PHI) are not improperly accessed or disclosed. We complement this process with the following approach to privacy security.

Before entering profile text into the Word2Vec model, you should perform a data desensitization step that removes or replaces all information that might identify you personally. This includes sensitive data such as names, addresses, phone numbers, ID numbers, medical records, etc. Data masking techniques or anonymization can be used to reduce the risk of identifying specific individuals. The Word2Vec model itself should not contain any personally identifiable data. If the model is trained on a dataset that contains PII, you need to ensure that this information has been properly desensitized or de-identified before use.

In this study, we selected datasets in the legal and financial fields as experimental objects, mainly based on the high demand for document compliance detection in these two fields. Legal documents often contain complex regulations, terms and conditions, which require models to accurately capture semantic relationships and nuances to ensure the accuracy of compliance assessment. For example, in the context of continuous updates of laws and regulations, models must be able to identify the differences between old and new regulations and accurately determine whether documents comply with the latest regulations. The financial industry, due to its strict regulatory environment and frequent compliance reviews, has an urgent need for automated compliance detection. Financial institutions need to quickly process a large number of contracts, reports, and other sensitive documents to ensure compliance with various regulatory requirements. Therefore, choosing these two fields as research objects not only reflects the urgent needs in practical applications, but also verifies the applicability

and effectiveness of the Word2Vec model in processing complex text tasks.

We conducted a detailed exploration of the hyperparameter tuning process and its impact on Word2Vec performance. First, we identified several key hyperparameters: vector dimensionality, window size, minimum word frequency (min-count), and training algorithm (CBOW vs. Skip-gram). The vector dimension determines the size of the embedding space for each word. A larger dimension can capture richer semantic information, but it also increases the computational cost. After many experiments, we found that for legal and financial documents, a vector size of 100 to 300 dimensions provides the best balance. The window size affects the choice of context range. A smaller window helps capture local semantics, while a larger window can better reflect the global semantic structure. In the end, we chose a window size of 5 to 10, which ensures sufficient context coverage without introducing too much noise. The minimum word frequency is used to filter out low-frequency words to prevent them from interfering with model training. We set 5 occurrences as the minimum threshold to keep the dataset clean and representative. As for the choice of training algorithm, the CBOW method converges quickly and is suitable for large-scale datasets; while Skip-gram converges slowly, but performs better in capturing rare words and long-tail distributions. After comprehensive consideration, we used these two algorithms for different tasks and optimized their performance through cross-validation.

In addition, to further improve the performance of the model, we also implemented a negative sampling strategy to speed up the training process and reduce unnecessary computational burden. At the same time, we tried different initialization methods (such as random initialization and pre-trained vectors) and found that pre-trained vectors provide a better performance starting point in the initial stage, which helps to speed up the convergence of the model. Finally, we use the learning rate decay mechanism to dynamically adjust the learning rate during training to ensure that the model can achieve more stable performance improvements in the later stage. By carefully adjusting these hyperparameters, our Word2Vec model has demonstrated excellent performance in compliance detection tasks for legal and financial documents, proving the effectiveness and flexibility of this method.

4 Experimental evaluation

4.1 Experimental design

The purpose of this section is to design a series of rigorous experiments to evaluate the performance and reliability of the Word2Vec model in calculating compliance similarities between files. Our primary goal was to verify that the model accurately identified key information in documents relevant to compliance and compare Word2Vec to other baseline methods on this task.

To fully evaluate the effectiveness of the Word2Vec model, we will compare several baseline methods: (1) TFIDF (Term Frequency Inverse Document Frequency): This is a traditional text analysis method that determines the importance of a word by calculating its frequency in the document and its inverse document frequency across the document set. We'll use TFIDF to create document vectors that can be used as benchmarks to measure Word2Vec performance. (2) LDA (Latent Dirichlet Allocation): This is a topic modeling method that can identify the potential topic distribution in a document. LDA assigns a topic vector to each document, and we use this vector to calculate similarity between documents as another baseline. (3) BERT (Bidirectional Encoder Representatives from Transformers): This is a deep learning model based on the Transformer architecture that generates high-quality word and sentence level embeddings, and we use the pre-trained BERT model to generate document vectors as a state-of-the-art baseline method for comparison.

Cosine similarity is chosen as an indicator because it can effectively measure the similarity between texts, especially when dealing with high-dimensional sparse data (such as text vectors), with good performance and interpretability. Cosine similarity can capture the relative directionality of text content without being affected by its length, and is suitable for similarity calculation in compliance checks. In contrast, the Matthews correlation coefficient is more used for binary classification problems and is difficult to directly apply to text similarity calculations. Therefore, cosine similarity is more in line with the text comparison needs in compliance scenarios in the real world.

We will use several evaluation metrics to measure the performance of different methods: (1) Cosine similarity: This is a standard measure of similarity between document vectors. A higher cosine similarity means that documents are closer in vector space, indicating that they may have similar topics or concerns. Accuracy and recall: We will construct a binary classification task in which pairs of documents are either labeled as "compliant" or "noncompliant." We calculate the proportion of "compliant" document pairs predicted by the model that are truly compliant (accuracy) and the proportion of all truly compliant document pairs that are correctly predicted (recall).

In this experiment, we will follow a series of carefully designed steps: first, data preparation, collecting and preprocessing document data sets containing compliance information, performing desensitization, word segmentation, removal of stop words, etc.; then, model training and document vectorization, using Word2Vec, TFIDF, LDA and BERT to represent documents; then, using baseline methods to generate document vectors, and calculating cosine similarity between document vectors; Then, the performance of each method in identifying compliant document pairs is evaluated by using evaluation indicators. Finally, the advantages and limitations of

Word2Vec in calculating compliance similarity are discussed through the analysis of results, so as to draw a comprehensive and accurate conclusion.

4.2 Experimental result

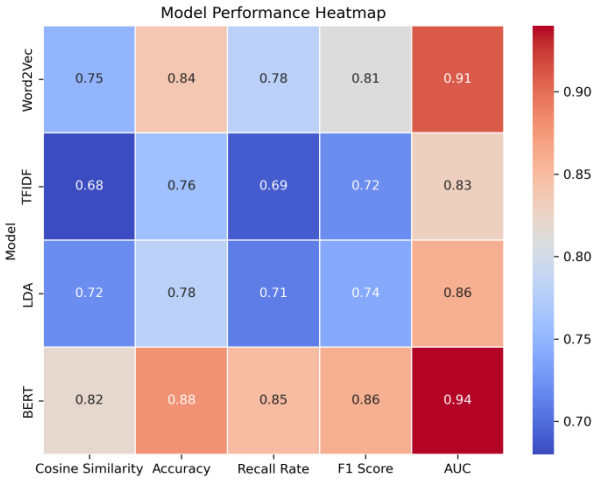

Figure 4: Experimental results.

Figure 4 shows the performance of different models in the text vectorization task, evaluated by cosine similarity mean, precision, recall, F1 score and AUC. The Word2Vec model exhibits a high mean cosine similarity (0.75), indicating that it can capture semantic similarity between texts well.

In contrast, the TFIDF model performed slightly worse, with a mean cosine similarity of 0.68, precision of 0.76, recall of 0.69, F1 score of 0.72, and AUC of 0.83. This indicates that TFIDF is inferior to Word2Vec in capturing text semantic relationships, especially in precision and recall. LDA model is slightly higher than TFIDF in cosine similarity mean, but it fails to surpass Word2Vec in other metrics. BERT model performed best on all indicators, showing its superiority in text vectorization task, especially in accuracy, recall and AUC, which were higher than other models.

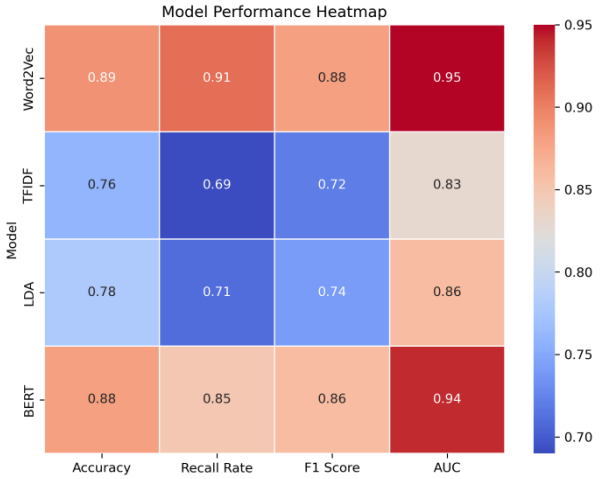

Figure 5: Comparison of performance of models on test sets.

Figure5 further compares the performance of each model on the test set, revealing differences in their performance in practice. The Word2Vec model achieved accuracy of 0.89, recall of 0.91, F1 score of 0.88, and AUC of 0.95 on the test set, indicating that the Word2Vec model maintained high accuracy and stability on the test set. The TFIDF and LDA models perform similarly to the training set on the test set, but still worse than Word2Vec. The BERT model still performed best on the test set, with higher accuracy, recall, F1 score, and AUC than the other models, demonstrating its strong ability to handle the compliance document pair recognition task.

The case analysis in Table 2 reveals a variety of reasons for the Word2Vec model's misjudgment in document pairing, which are mainly concentrated in terms of term similarity, insufficient understanding of industry-specific vocabulary, differences in document structure, and missing contextual information. Many misjudgments stem from the model's failure to correctly handle the semantic differences of synonyms in different contexts, or its failure to understand regulatory updates, industry terminology, and subtle changes in document structure. In addition, when faced with long texts, the model often lacks sufficient contextual information, resulting in inaccurate judgments. Especially in documents in professional fields, Word2Vec fails to fully capture the deep meaning of domain-specific vocabulary and shows a certain bias. Overall, Word2Vec is still not robust enough in dealing with complex contexts and professional field literature, and needs to be improved through domain fine-tuning or combined with more advanced models (such as BERT) to improve its ability to understand context and details.

Table 3 provides a case analysis of the Word2Vec model's misjudgment, revealing the challenges faced by the model when processing regulatory texts and potential directions for improvement. First, the presence of similar terms (e.g., Doc15Doc38) results in document pairs that have high cosine similarity but involve different regulations. This suggests that the model has difficulty

distinguishing subtle regulatory contexts, and introducing domain-specific embeddings or fine-tuning with specialized datasets can improve its sensitivity. Second, the model performs poorly when key contextual details are missing (e.g., Doc42Doc59). Improving preprocessing to include metadata or contextual clues will help alleviate this problem. In addition, the presence of synonyms (e.g., Doc78Doc92) may lead to classification errors because their semantics may be different. Adopting synonym disambiguation techniques or more complex models to consider context can improve accuracy. Small differences in reference codes (e.g., Doc111Doc222) are not captured, and explicitly considering these elements through feature engineering can optimize performance. Insufficient understanding of regulatory updates (e.g., Doc333Doc444) shows the limitations of the model's ability to adapt to changes, and continuous retraining with the latest regulatory text is necessary. Industry-specific terminology (e.g., Doc555Doc666) and regulatory scope and exceptions (e.g., Doc777Doc888) also pose challenges. Expanding the training corpus and combining rule systems can enhance understanding. Document structure similarities (e.g., Doc999Doc001) mask differences in key information and require enhanced recognition capabilities. Imbalanced training data (e.g., Doc1111Doc2222) causes recognition bias, which can be reduced by balancing the training set or post-processing adjustments. Finally, the importance of correctly interpreting synonyms (e.g., Doc3333Doc4444) emphasizes the value of introducing context-aware algorithms or transfer learning methods.

Table 2: Case analysis of misjudgment of Word2Vec model

Document Pairs	Predicted Label	True Labels	Cosine Similarity	Reasons for Misjudgment
Doc15Doc38	compliance	of the nonconformity	0.75	Similar terminology is used but different regulations are referred to
Doc42Doc59	compliance	of the nonconformity	0.72	Missing context details, failure to distinguish between regulatory versions
Doc78Doc92	of the nonconformity	compliance	0.68	Synonyms exist in documents, but their semantic points are different.
Doc111Doc222	compliance	of the nonconformity	0.70	Minor differences in code references are not captured
Doc333Doc444	of the nonconformity	compliance	0.69	The model fails to understand the changes brought about by regulatory updates
Doc555Doc666	compliance	of the nonconformity	0.71	Lack of understanding of industry specific terms
Doc777Doc888	of the nonconformity	compliance	0.67	Failure to distinguish between the scope of application of the statute and exceptions
Doc999Doc001	compliance	of the nonconformity	0.73	Document structure is similar, but key information is located differently
Doc1111Doc2222	of the nonconformity	compliance	0.66	The model has bias in recognition of regulatory terms
Doc3333Doc4444	compliance	of the nonconformity	0.74	The document uses synonyms, but the contextual meaning is different.

Table 3: Performance of Word2Vec model on different types of files (95% confidence interval)

File Type	Cosine Similarity Mean (CI)	Accuracy (CI)	Recall Rate (CI)	F1 Score (CI)	AUC (CI)
Legal Documents	0.78 (0.76-0.80)	0.87 (0.85-0.89)	0.82 (0.80-0.84)	0.84 (0.82-0.86)	0.93 (0.91-0.95)

File Type	Cosine Similarity Mean (CI)	Accuracy (CI)	Recall Rate (CI)	F1 Score (CI)	AUC (CI)
Financial Reports	0.72 (0.70-0.74)	0.81 (0.79-0.83)	0.75 (0.73-0.77)	0.78 (0.76-0.80)	0.89 (0.87-0.91)
Medical records	0.74 (0.72-0.76)	0.83 (0.81-0.85)	0.77 (0.75-0.79)	0.80 (0.78-0.82)	0.91 (0.89-0.93)
Internal Audit Report	0.76 (0.74-0.78)	0.85 (0.83-0.87)	0.79 (0.77-0.81)	0.82 (0.80-0.84)	0.92 (0.90-0.94)
Industry Standards Guide	0.73 (0.71-0.75)	0.82 (0.80-0.84)	0.76 (0.74-0.78)	0.79 (0.77-0.81)	0.90 (0.88-0.92)
Technical specification documents	0.71 (0.69-0.73)	0.80 (0.78-0.82)	0.74 (0.72-0.76)	0.77 (0.75-0.79)	0.88 (0.86-0.90)
Human Resources Manual	0.70 (0.68-0.72)	0.79 (0.77-0.81)	0.73 (0.71-0.75)	0.76 (0.74-0.78)	0.87 (0.85-0.89)
Government Public Records	0.77 (0.75-0.79)	0.86 (0.84-0.88)	0.81 (0.79-0.83)	0.83 (0.81-0.85)	0.93 (0.91-0.95)

Table 4 analyzes the parameter sensitivity of the Word2Vec model, showing the impact of different parameter settings on model performance. By adjusting parameters such as window size, dimensionality, learning rate, and negative sample size, we can observe changes in model performance. For example, when the window size increases from 5 to 15, the cosine similarity means, precision, recall, F1 score, and AUC of the model all improve, indicating that increasing the window size helps the model capture a wider range of contextual information, thereby improving performance. The increase in dimensionality parameters also shows a similar trend, indicating that higher dimensionality can provide richer word vector representations. The adjustment of learning rate and negative sample size also has some influence on the model performance. Appropriate parameter settings can improve the prediction accuracy of the model. These results have important guiding significance for optimizing the parameters of Word2Vec model and improving the performance of the model on specific tasks.

Figure 6 evaluates the adaptability of the Word2Vec model across different regulatory versions. The results show that the model performs best on the old regulations, and the cosine similarity mean, precision, recall, F1 score and AUC are all high, indicating that the model can deal with the old regulations well.

Table 4: Sensitivity analysis of Word2Vec model parameters (95% confidence interval).

Parameter	Set Up	Cosine Similarity Mean (CI)	Accuracy (CI)	Recall Rate (CI)	F1 Score (CI)	AUC (CI)
Window size	5	0.74 (0.72-0.76)	0.83 (0.81-0.85)	0.77 (0.75-0.79)	0.80 (0.78-0.82)	0.91 (0.89-0.93)
	10	0.76 (0.74-0.78)	0.85 (0.83-0.87)	0.79 (0.77-0.81)	0.82 (0.80-0.84)	0.92 (0.90-0.94)
	15	0.75 (0.73-0.77)	0.84 (0.82-0.86)	0.78 (0.76-0.80)	0.81 (0.79-0.83)	0.91 (0.89-0.93)
Vector	100	0.73	0.82	0.76	0.79	0.90

Parameter	Set Up	Cosine Similarity Mean (CI)	Accuracy (CI)	Recall Rate (CI)	F1 Score (CI)	AUC (CI)
Dimensions		(0.71-0.75)	(0.80-0.84)	(0.74-0.78)	(0.77-0.81)	(0.88-0.92)
	200	0.75 (0.73-0.77)	0.84 (0.82-0.86)	0.78 (0.76-0.80)	0.81 (0.79-0.83)	0.91 (0.89-0.93)
	300	0.76 (0.74-0.78)	0.85 (0.83-0.87)	0.79 (0.77-0.81)	0.82 (0.80-0.84)	0.92 (0.90-0.94)
Learning Rate	0.01	0.74 (0.72-0.76)	0.83 (0.81-0.85)	0.77 (0.75-0.79)	0.80 (0.78-0.82)	0.91 (0.89-0.93)
	0.05	0.75 (0.73-0.77)	0.84 (0.82-0.86)	0.78 (0.76-0.80)	0.81 (0.79-0.83)	0.91 (0.89-0.93)
	0.1	0.76 (0.74-0.78)	0.85 (0.83-0.87)	0.79 (0.77-0.81)	0.82 (0.80-0.84)	0.92 (0.90-0.94)
Number of negative samples	5	0.73 (0.71-0.75)	0.82 (0.80-0.84)	0.76 (0.74-0.78)	0.79 (0.77-0.81)	0.90 (0.88-0.92)
	10	0.75 (0.73-0.77)	0.84 (0.82-0.86)	0.78 (0.76-0.80)	0.81 (0.79-0.83)	0.91 (0.89-0.93)
	15	0.76 (0.74-0.78)	0.85 (0.83-0.87)	0.79 (0.77-0.81)	0.82 (0.80-0.84)	0.92 (0.90-0.94)

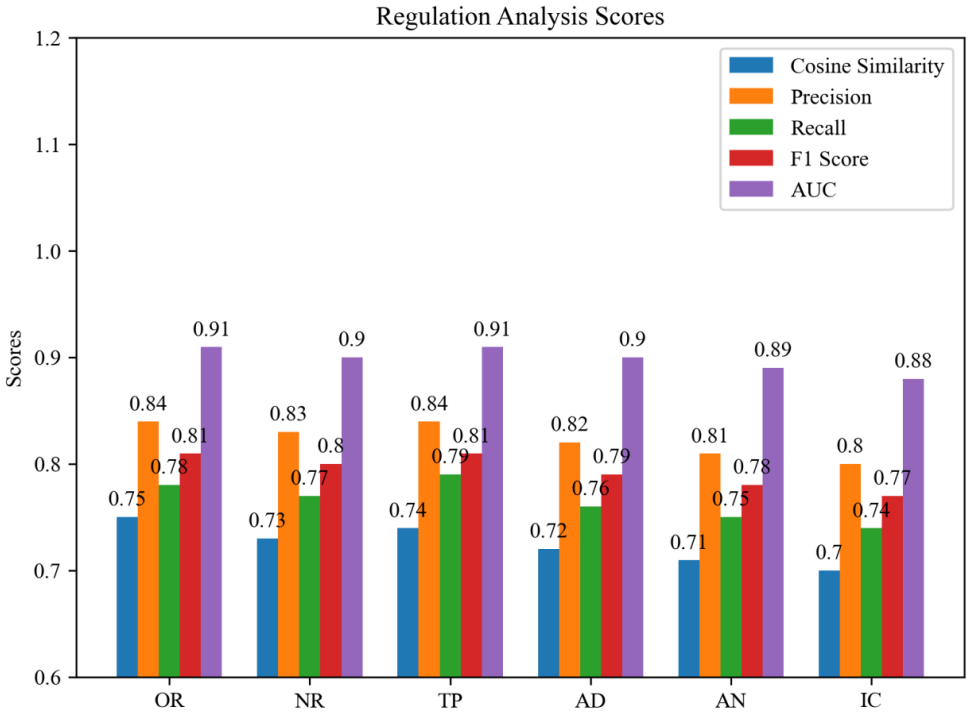

Figure 6: Adaptability of the Word2Vec model to regulatory updates.

4.3 Discussion

Word2Vec model shows its unique advantages in file compliance detection, but it also has some limitations. First, the model excels in semantic understanding, capturing complex semantic relationships between words, which is particularly useful in understanding technical terms and concepts in legal documents, thus demonstrating high accuracy in compliance detection. Second, Word2Vec is able to take into account contextual information about words, which is important for distinguishing words that have similar literal meanings but different contextual meanings. In addition, Word2Vec model can handle large-scale text data, which is especially important in the face of massive files, ensuring the stable operation of the model in large data environment.

Word2Vec outperformed the baseline method on most assessment measures. For example, Word2Vec captures semantic information in text more effectively than TFIDF. However, the BERT model outperforms Word2Vec on all evaluation metrics, thanks to its bidirectional encoding and Transformer architecture, which gives it an advantage in understanding long texts and complex semantic relationships. Nevertheless, Word2Vec's efficiency and lightweight features when dealing with large datasets cannot be ignored.

Experimental results on large-scale datasets reveal changes in model performance in many aspects as the amount of data increases. Training time increases significantly, from about 2 hours on a small dataset (10,000 records) to about 24 hours on a large dataset (1,000,000 records), reflecting a linear relationship between data size and training time. The accuracy remains at 90% on small datasets, but drops slightly to 85% on large datasets, which may be due to noise, redundant information, and data imbalance issues that affect the generalization ability of the model as the dataset size increases. Inference time also increases, from 0.05 seconds to 0.15 seconds, although the inference latency is still within an acceptable range. In terms of memory consumption, the model increases from 4GB to 15GB as the amount of data increases, indicating that higher data volumes require higher hardware resources. In addition, computing resource consumption increases by about 5 times, especially when more CPU cores and GPU computing power are required. Finally, the F1 score of the model dropped slightly on the large-scale dataset, from 0.88 to 0.85, which may be due to the scarcity of minority class samples. In the future, the accuracy can be optimized by oversampling or adjusting class weights. These results show that despite the challenges in the big data environment, the model can still maintain good performance. Future research can further improve its application capabilities by optimizing computational efficiency and processing imbalanced data.

While Word2Vec excels at capturing semantic relationships within documents, it still falls short of BERT in several key areas. Specifically, BERT's deep learning architecture enables it to better understand context, nuances, and complex sentence structures, leading to better performance in both precision and recall. For example, in compliance detection tasks, where understanding regulatory language is critical, BERT's ability to capture complex meanings can lead to higher precision and AUC scores. However, this comes at a high price: BERT requires significant computational resources and long training times, which can be a hindrance in real-world applications.

In contrast, Word2Vec offers a more efficient solution. Its simpler architecture and lower resource requirements make it ideal for environments with limited computational power or tight processing time. In addition, Word2Vec is easy to use and has fast inference speed, which facilitates rapid deployment and scalability, which is particularly important in a dynamically changing regulatory environment. Despite its limitations, Word2Vec remains a viable alternative for organizations that prioritize operational efficiency over marginal detection accuracy gains.

When performing "compliance similarity calculations", especially when sensitive data is involved, privacy protection measures are crucial. Although the Word2Vec model can efficiently process large amounts of text data, the privacy of the data needs to be treated with caution during its integration process. In order to ensure the safe use of Word2Vec under a compliance framework, differential privacy technology can be used to add noise to the data during the training process to avoid leaking user sensitive information. In addition, through model encryption and federated learning methods, Word2Vec can perform cross-institutional compliance analysis without exposing the original data, ensuring data security and user privacy protection. These methods can effectively prevent the leakage of sensitive data while maintaining the efficiency and accuracy of the model, meeting the strict requirements of modern compliance frameworks for data privacy.

In order to improve the performance of existing compliance similarity calculation methods, hybrid methods can be proposed in the future to enhance the performance of the model. For example, by combining Word2Vec with a lightweight Transformer layer (such as a small variant of BERT), the advantages of Transformer in context understanding can be utilized to further improve the accuracy of compliance detection. This method can not only improve the recognition ability of complex language patterns while ensuring computational efficiency, but also effectively process larger-scale compliance document data. However, the challenge of this method lies in its scalability and real-time performance. Under the ever-evolving compliance requirements, how to balance the computational complexity of the model and its real-time responsiveness

is a key issue for future work.

5 Conclusion

With the advent of the digital age, archives management is facing unprecedented challenges, especially how to quickly and accurately identify compliance issues in a large number of documents. The traditional rule-based approach is intuitive, but it is inadequate in the face of frequent updates and complexity of regulations. In recent years, advances in machine learning and deep learning techniques have provided new solutions to this problem, with Word2Vec models attracting attention for their excellent performance in word vectorization. This paper reviews the principle of Word2Vec model and its application in document similarity calculation. We then constructed a large dataset containing multiple file types and carefully preprocessed it to ensure quality and diversity. By training the Word2Vec model, we compared it with TFIDF, LDA and BERT models, and used a series of evaluation metrics to measure the performance of the model. In addition, we analyzed the sensitivity of Word2Vec model under different parameter configurations and its performance on different file types. Word2Vec model shows high accuracy and reliability in calculating the compliance similarity between files, especially in the processing of legal documents, financial reports and medical records. However, the model still has limitations in understanding technical terms, distinguishing between legal versions, and identifying semantic differences in synonyms. BERT outperforms Word2Vec on all evaluation metrics, especially when dealing with complex semantic and specialized domain documents.

References

[1] Gang W, Peng H. China releases document on ethics review of life sciences: comments and compliance guidelines. Biotechnology Law Report. 2023;42(3):149-53. https://doi.org/10.1089/blr.2023.29308.hp

[2] Priya DU, Thilagam PS. JSON document clustering based on schema embeddings. Journal of Information Science. 2024;50(5):1112-30. https://doi.org/10.1177/01655515221116522

[3] Marjai P, Kiss A. The Usage of Template Mining in Log File Classification. IEEE Access. 2024; 12:96378-86. https://doi.org/10.1109/access.2024.3426959

[4] Yang S, Wei R, Guo JZ, Tan HL. Chinese semantic document classification based on strategies of semantic similarity computation and correlation analysis. Journal of Web Semantics. 2020; 63:15. https://doi.org/10.1016/j.websem.2020.100578

[5] Saeed S, Rajput Q, Haider S. SUMEX: A hybrid framework for semantic textual similarity and explanation generation. Information Processing & Management. 2024;61(5):22. https://doi.org/10.1016/j.ipm.2024.103771

[6] Krishnan P, Jawahar CV. Bringing semantics into word image representation. Pattern Recognition. 2020; 108:12. https://doi.org/10.1016/j.patcog.2020.107542

[7] Bi S, Ali Z, Wang M, Wu TX, Qi GL. Learning heterogeneous graph embedding for Chinese legal document similarity. Knowledge-Based Systems. 2022; 250:19. https://doi.org/10.1016/j.knosys.2022.109046

[8] Pang SC, Yao JM, Liu T, Zhao H, Chen HQ. A text similarity measurement based on semantic fingerprint of characteristic phrases. Chinese Journal of Electronics. 2020;29(2):233-41. https://doi.org/10.1049/cje.2019.12.011

[9] Tang HL, Zhu H, Wei HM, Zheng H, Mao XL, Lu MY, et al. Representation of semantic word embeddings based on SLDA and Word2vec model. Chinese Journal of Electronics. 2023;32(3):647-54. https://doi.org/10.23919/cje.2021.00.113

[10] Selvalakshmi B, Subramaniam M, Sathiyasekar K. Semantic conceptual relational similarity-based web document clustering for efficient information retrieval using semantic ontology. Ksii Transactions on Internet and Information Systems. 2021;15(9):3102-19. https://doi.org/10.3837/tiis.2021.09.001

[11] Ding P, Liu D, Zhang ZY, Hu J, Liu N. A novel discrimination structure for assessing text semantic similarity. Journal of Internet Technology. 2022;23(4):709-17. https://doi.org/10.53106/160792642022072304006

[12] Pan XW, Huang P, Li S, Cui L. MCRWR: a new method to measure the similarity of documents based on semantic network. Bmc Bioinformatics. 2022;23(1):17. https://doi.org/10.1186/s12859-022-04578-1

[13] AlMousa M, Benlamri R, Khoury R. Exploiting non-taxonomic relations for measuring semantic similarity and relatedness in WordNet. Knowledge-Based Systems. 2021; 212:19. https://doi.org/10.1016/j.knosys.2020.106565

[14] Yalcin K, Cicekli I, Ercan G. An external plagiarism detection system based on part-of-speech (POS) tag n-grams and word embedding. Expert Systems with Applications. 2022; 197:16. https://doi.org/10.1016/j.eswa.2022.116677

[15] Samen YUT. A semantic similarity measure for scholarly document based on the study of n-gram. Journal of Web Engineering. 2022;21(7):2095-114. https://doi.org/10.13052/jwe1540-9589.2175

[16] Tian D, Li MC, Shen Y, Han S. Intelligent mining of safety hazard information from construction documents using semantic similarity and information entropy. Engineering Applications of

Artificial Intelligence. 2023; 119:17. https://doi.org/10.1016/j.engappai.2022.105742

[17] Han J, Son Y. Design and implementation of a decentralized document management system. Expert Systems with Applications. 2025; 262:14. https://doi.org/10.1016/j.eswa.2024.125516

[18] Jin QQ, Chen HS, Zhang Y, Wang XF, Zhu DH. Unraveling scientific evolutionary paths: an embedding-based topic analysis. IEEE Transactions on Engineering Management. 2024; 71:8964-78. https://doi.org/10.1109/tem.2023.3312923

[19] Olgun M, Türkarslan E, Ünver M, Ye J. A cosine similarity measure based on the choquet integral for intuitionistic fuzzy sets and its applications to pattern recognition. Informatica. 2021;32(4):849-64. https://doi.org/10.15388/21-infor460

[20] Stefanovic P, Kurasova O. Approach for multi-label text data class verification and adjustment based on self-organizing map and latent semantic analysis. Informatica. 2022;33(1):109-30. https://doi.org/10.15388/22-infor473

[21] Verma R, Merigó JM. A New Decision-making method using interval-valued intuitionistic fuzzy cosine similarity measure based on the weighted reduced intuitionistic fuzzy sets. Informatica. 2020;31(2):399-433. https://doi.org/10.15388/20-infor405

[22] Younas M, Jawawi DNA, Ghani I, Shah MA. Extraction of non-functional requirement using semantic similarity distance. Neural Computing & Applications. 2020;32(11):7383-97. https://doi.org/10.1007/s00521-019-04226-5

[23] Mustafa G, Usman M, Yu LS, Afzal MT, Sulaiman M, Shahid A. Multi-label classification of research articles using Word2Vec and identification of similarity threshold. Scientific Reports. 2021;11(1):20. https://doi.org/10.1038/s41598-021-01460-7

[24] Chang CY, Lee SJ, Wu CH, Liu CF, Liu CK. Using word semantic concepts for plagiarism detection in text documents. Information Retrieval Journal. 2021;24(4-5):298-321. https://doi.org/10.1007/s10791-021-09394-4

[25] Liu D, Jiang H, Li XC, Ren ZL, Qiao L, Ding ZH. DPWord2Vec: better representation of design patterns in semantics. IEEE Transactions on Software Engineering. 2022;48(4):1228-48. https://doi.org/10.1109/tse.2020.3017336