Ensemble Methods for Word Embedding Model Based on Judicial Text

Chunyu Xia, Tieke He(✉), Jiabing Wan, and Hui Wang

State Key Laboratory for Novel Software Technology,
Nanjing University, Nanjing 210093, China
hetieke@gmail.com

Abstract. With the continuous expansion of computer applications, scenarios such as machine translation, speech recognition, and message retrieval depend on the techniques of the natural language processing. As a technique for training word vectors, Word2vec is widely used because it can train word embedding model based on corpus and represent the sentences as vectors according to the training model. However, as an unsupervised learning model, word embedding can only characterize the internal relevance of natural language in non-specific scenarios. For a specific field like judicial, the method of expanding the vector space by creating a professional judicial corpus to enhance the accuracy of similarity calculation is not obvious, and this method is unable to provide further analysis for similarity in cases belonging to the same type. Therefore, based on the original word embedding model, we extract factors such as fines and prison term to help identify the differences, and attach the label of the case to complete supervised ensemble learning. The result of the ensemble model is better than any result of single model in terms of distinguishing whether they are the same type. The experimental result also reveal that the ensemble method can effectively tell the difference between similar cases, and is less sensitive to the details of the training data, the choice of training plan and the contingency of a single inaccurate training run.

Keywords: Natural language · Word embedding · Judicial text · Ensemble learning · Supervised learning

1 Introduction

In the past decade, artificial intelligence (AI) has become a popular term that covers all advanced smart technologies such as adversarial competition, computer vision and natural language processing, bringing amazing advances to computer-aided intelligence. As an extension of AI, Natural Language Processing (NLP) has achieved unprecedented success in dealing with machine translation, speech recognition, information retrieval and text similarity. Although NLP has evolved into different aspects of a smart society, intelligent justice lacks specific and accurate applications.

W. Ni et al. (Eds.): WISA 2019, LNCS 11817, pp. 309–318, 2019.
https://doi.org/10.1007/978-3-030-30952-7_31

In terms of intelligent justice, NLP can help the court save time by identifying similar cases and recommending relevant text [1]. The word embedding method overcomes the basic problem in the judicial text, namely natural semantics [13]. In addition, Word2vec-based technology can help with word embedding in a more efficient manner. Unfortunately, word embedding lacks the ability to overcome common pitfalls in natural language, such as polyphonic words. These common deficiencies can lead to accidental deviations, and accidental deviations should be eliminated in the field of judicial texts.

In contrast, ensemble methods for embedding words can help reduce accidental mistakes. Through experiments, we found that the set model performed well in terms of abnormal similarity due to inaccurate expression. More importantly, this model can be induced into a dimensionality reduction method. Projecting a high-dimensional vector into a three-dimensional vector simplifies calculations and shows excellent data fit.

2 Related Work

Looking back at the history of text similarity architectures, the classical text similarity architecture is based on specific differences between similar sentences. The definition of classical difference is the difference in the number and length of different words represented by two sentences, or in short, the ratio of intersection to union. Due to the rigid model and limited vocabulary, this classic building was replaced by a high-dimensional model based on a large corpus. In view of this model, the term frequency-inverse document frequency (TF-IDF) and bag of words (BOW) appeared. Their main idea is that if you create a corpus full of words, you can project each sentence into a high-dimensional vector. However, creating a corpus can only convert sentences into vectors without considering word order or synonyms. This problem actually has a fundamental impact on the emergence of word embedding.

Word embedding is a concept that describes the relationship between adjacent words and tends to predict the meaning of word. For example, a regular sentence can be converted into an inverted sentence, or even another sentence with completely different words can be reconstructed, albeit with the same meaning. Based on classical models or corpus models, it is hard to distinguish similarities and even categorize them as opposite vectors by mistake. Nevertheless, the word embedding can simulate real natural language scenes and produce mutually replaceable words.

In addition, in 2013, Mikolov proposed a new technique to train word embedding, which is called Word2vec [12]. Word2vec is a method of constructing a word embedding model after text training, with the options for continuous bag-of-words (CBOW) and skip-gram (SG) [10]. With the help of Word2vec, the concept of word embedding can be extended into other applications besides the NLP domain. Recent applications have trained word embedding of user actions like clicks, requests and searches to provide personal recommendations. Domains like E-commerce, E-business, and Market have utilized this approach to handle search rankings [4].

In terms of judicial text [5], in order to enhance the scalability of word embedding, relative research has added a professional corpus to trained word embedding. This method is to refine the word embedding model and improve the accuracy of comparing similar sentences. Although the vector space seems to work well in experiments due to the expansion of professional corpus, this solution is very poor when dealing with totally different categories of cases. The similarities in different cases may be high, not down to zero. More importantly, if the two cases are of the same type, the exact difference cannot be accurately stated.

3 Model Design

In order to avoid the shortcomings of word embedding, we need to convert unsupervised learning into supervised learning [2]. Word2vec is an unsupervised learning without tags [9]. An obvious problem with Word2vec is that the training model is only trying to cluster data features in the subconscious. Sometimes, when coincident feature learning happens to fit the essential differences in the category, the model can produce better results. More often than not, training models simply over-fitting features and creating some unreasonable boundaries. Therefore, we use supervised learning to address these shortcomings.

It is undeniable that we cannot supervise the word embedded model during Word2vec training. Instead, we can consider the original model as part of the new supervisory training model [14]. As a principle feature, fines and imprisonment are also important factors in dealing with similarities.

In this paper, we present an ensemble method for offsetting defects displayed in word embedding. The general idea is that we can use Word2vec to train word embedding in judicial texts and then combine the similarities of Word2vec calculations with other features [15]. By adjusting these features, we can adapt to the characteristics of different types of cases.

As described in the image, the entire process is derived from the cail data set. The first operation is to extract a single type of useful data from a composite data set, such as text words from facts, penalties from money, and prison terms from imprisonment. Once the data is ready, we can define different operations for them. In particular, the operation of text words is exactly the same as in the normal Word2vec model, and it is optional to train word embedding based on professional corpora. However, the most significant difference is the vector formed by the regular Word2vec model and other gaps, which are calculated by comparing the corresponding penalty and prison terminology [3]. The accusations in the figure reveal the basic idea that we can transform an unsupervised model into a supervised model. Ultimately, the entire data and tags can be injected into the deep neural network (DNN) model (Figs. 1 and 2).

DNN is an effective machine learning method that facilitates data fitting [6]. The ensemble DNN model we propose here is a DNN model that absorbs the advantages of unsupervised learning and supervised learning. By modifying the parameters that represent weights in specific situations, we can modify the

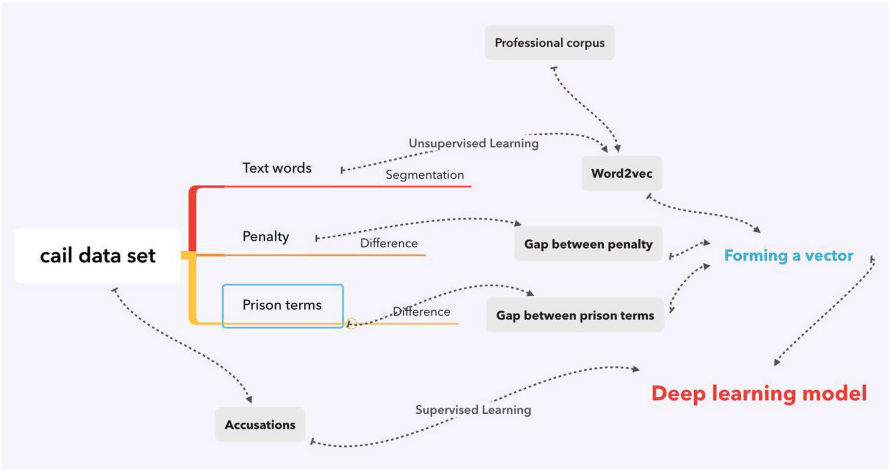

Fig. 1. Model architecture

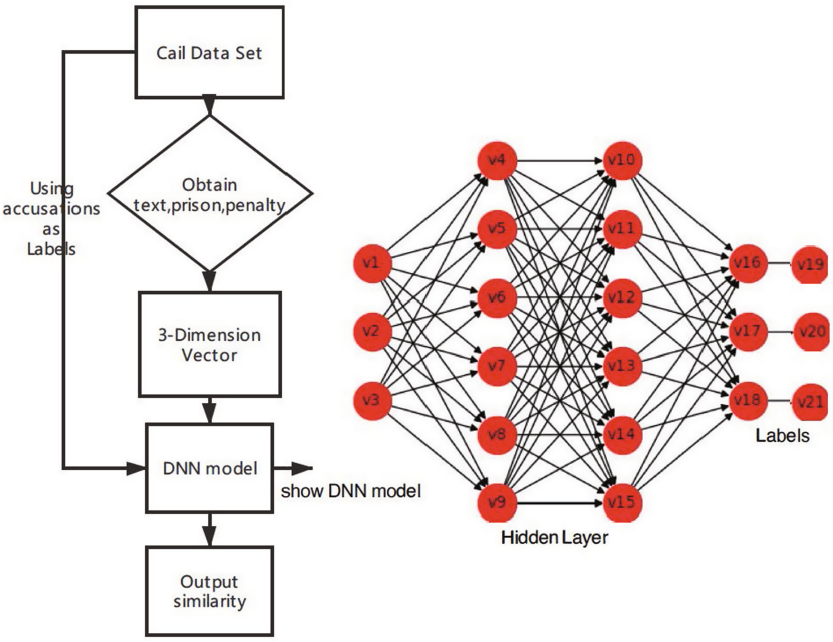

Fig. 2. Ensemble DNN model

model to accommodate minor changes. On the one hand, unsupervised learning can derive some features from a general point of view, which may be somewhat inaccurate but correspond to natural language. On the other hand, supervised learning can compensate for shortcomings by precise gradient descent and data fitting. In short, the model designed here takes into account the particularity of the similarity of judicial texts.

4 Evaluation and Results

When evaluating this model, we applied this model to two extreme cases and general cases. By experimenting with these situations, we can make some conclusions based on the results.

4.1 Case 1: Texts of Different Types with High Similarity

For instance, theft cases are very similar to burglary cases except one is on the premise of breaking into the room. However, we observed that most burglary cases have a prison term of more than 10 years, so the gap between the same burglary cases is relatively small. In this case, we can increase the weight of imprisonment and the results are obvious. Even if the text is linguistically similar, the gap between prison terms can seriously affect similarity, and the end result will decline.

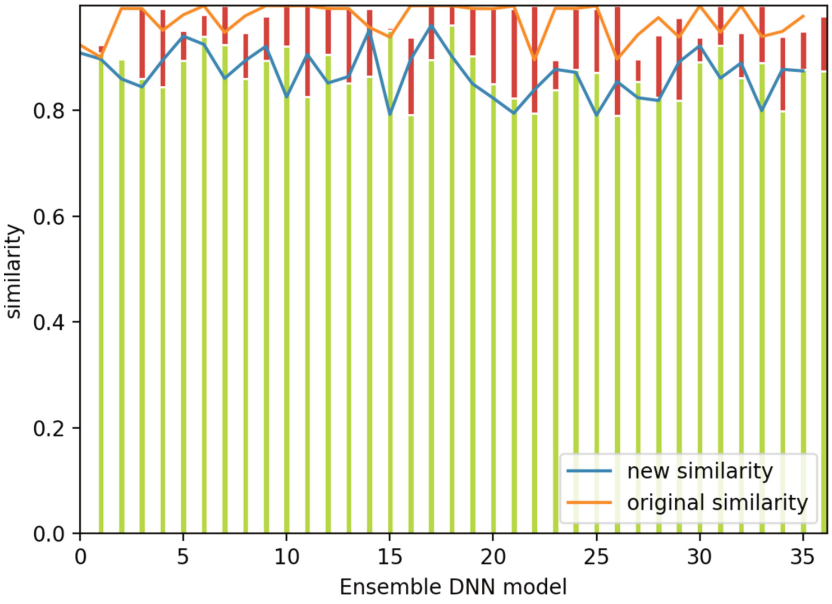

Fig. 3. Case 1 experiment (Color figure online)

We use a simple chart to illustrate this situation. The blue curve has the same trend as the orange curve. The consistency of the two curves means that the ensemble DNN model can be adapted to the original word embedding model and improved by means of other factors. More importantly, the green column is generally lower than the red column, which represents the effectiveness of the new model when dealing with confusing text (Table 1).

Table 1. Reduce the similarity of different types of text

	Low	High	Average
origin	0.91	0.99	0.96
ensemble	0.79	0.94	0.85

For cases with very similar categories, the initial similarity is very high, from the lowest value, the highest value and the average value. This is a highly error-prone situation. Based on this problem, our ensemble model can reduce the similarity in this case, and thus ensure the accuracy of the application in the actual scene (Fig. 3).

4.2 Case 2: Texts with Relatively Low Similarity but Belonging to the Same Type

In everyday life, it seems difficult to find sentences that are semantically different but expressing the same meaning. However, in the judicial text, this incident has taken place a lot. We can also take theft as an example. As a selected sample in the case of theft, the similarity between them is generally less than 0.8, which is a relatively low similarity. However, we found that all samples had less penalties and shorter prison terms. Because of this feature, we use the ensemble DNN model to train the data.

As is shown in the picture, the apparent result is that original similarities have been raised to 0.99 (shown in red columns). As I said above, the second situation is extremely special and thus the result can tell us the strength of this model to some degree. Even though these samples are not highly similar, the similarity grow rapidly combined with the weights of fines and prison terms. Therefore, it is safe to regard this model as a method to offset the unsupervised word embedding defects (Table 2).

As the figure shows, the obvious result is that the original similarity has been increased to 0.99 (shown in red columns). As mentioned above, the second case is very special, so the results can tell us the characteristics of this model to some extent. Even though these samples are not very similar, the similarities increase rapidly with the weight of fines and imprisonment. Therefore, it is safe to consider this model as a way to offset unsupervised word embedding defects (Fig. 4).

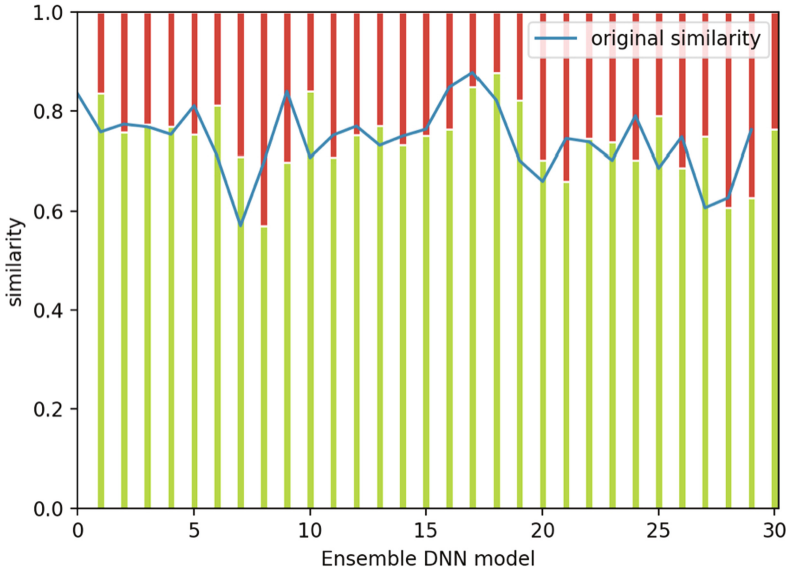

Fig. 4. Case 2 experiment (Color figure online)

Even in the same type of cases, occasionally the similarity is too low. The average similarity is only 0.78. In this case, we increased the similarity according to the prison terms and the penalty, and the final similarity can be increased to 0.97, which is regarded as a significant increase of 0.19.

Table 2. Raise the similarity of the same type of text

	Low	High	Average
origin	0.58	0.86	0.78
ensemble	0.94	0.99	0.97

4.3 General Case: High-Dimensional Space Mapping for Three-Dimensional Space

In addition to two extremely special cases, the normal situation of the model is the clustering problem in three-dimensional space. The general advantage of this model is dimensionality reduction. Word2vec represents a word embedding model in high-dimensional space, and the difference in high-dimensional space has been converted to a number between 0 and 1. This process is the first dimension reduction we know in the word embedding model (Fig. 5).

Besides, the overall DNN model combines this number with the gaps in other factors. This process also reduces complexity. The three-dimensional vector constitutes the input to the DNN model. By training this model, we can get the corresponding output to cluster relative types [7].

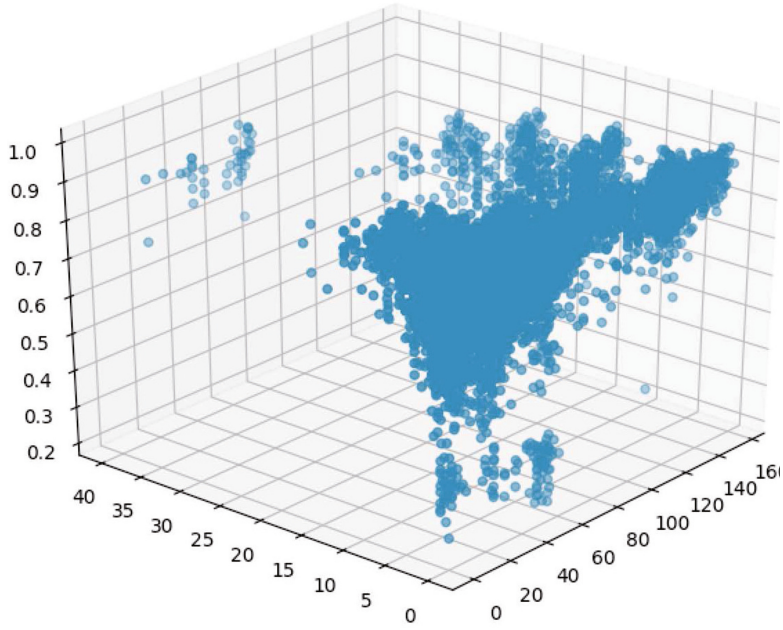

Fig. 5. General case experiment

Overall, for Case 1, we improved the accuracy of 0.29 because we increased the weight of the non-text indicators. For case 2, we only improved the accuracy of 0.12 due to its relatively high accuracy (Table 3).

Table 3. General increased accuracy

	Case1	Case2
origin-acc	0.65	0.86
ensemble-acc	0.94	0.98
variation	0.29	0.12

5 Discussion and Future Work

The ensemble method we propose here for word embedding is actually a way to reduce complexity and increase data fit. This model may be helpful in reducing accidental natural language similarities and enhancing the determinacy of reliability factors. The results also showed better applicability when dealing with sudden similar inaccuracies.

In terms of shortcomings, this is merely an attempt based on an ensemble approach that lacks sufficient reliability. The reconstructed vector is in three dimensions, and this reduced complexity may result in over-fitting of the data.

In addition, DNN models are sometimes less efficient, and several changes in parameters can have unpredictable effects.

For future work, when deciding the similarity of judicial texts, more attention should be paid to collecting useful factors [8]. We should consider choosing a more compatible model [11]. The new model should measure the importance of the different factors themselves during the training process. In addition, the stability of the new model is included when dealing with subtle changed parameters.

Acknowledgment. The work is supported in part by the National Key Research and Development Program of China (2016YFC0800805) and the National Natural Science Foundation of China (61772014).

References

1. Al-Kofahi, K., Jackson, P., Travers, T.E., Tyrell, A.: Systems, methods, and software for classifying text from judicial opinions and other documents, uS Patent 7,062,498, June 2006
2. Barlow, H.B.: Unsupervised learning. Neural Comput. **1**(3), 295–311 (1989)
3. Deng, L., Platt, J.C.: Ensemble deep learning for speech recognition. In: Fifteenth Annual Conference of the International Speech Communication Association (2014)
4. Grbovic, M., Cheng, H.: Real-time personalization using embeddings for search ranking at Airbnb. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 311–320. ACM (2018)
5. He, T., Lian, H., Qin, Z., Zou, Z., Luo, B.: Word embedding based document similarity for the inferring of penalty. In: Meng, X., Li, R., Wang, K., Niu, B., Wang, X., Zhao, G. (eds.) WISA 2018. LNCS, vol. 11242, pp. 240–251. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-02934-0_22
6. Hinton, G., et al.: Deep neural networks for acoustic modeling in speech recognition. IEEE Signal Process. Magaz. **29**, 82 (2012)
7. Iwayama, M., Tokunaga, T.: Cluster-based text categorization: a comparison of category search strategies. In: Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 273–280. Citeseer (1995)
8. Jaderberg, M., Vedaldi, A., Zisserman, A.: Deep features for text spotting. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8692, pp. 512–528. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10593-2_34
9. Jain, A.K., Farrokhnia, F.: Unsupervised texture segmentation using gabor filters. Pattern Recognit. **24**(12), 1167–1186 (1991)
10. Levy, O., Goldberg, Y.: Neural word embedding as implicit matrix factorization. In: Advances in Neural Information Processing Systems, pp. 2177–2185 (2014)
11. Marti, U.V., Bunke, H.: Using a statistical language model to improve the performance of an hmm-based cursive handwriting recognition system. In: Hidden Markov Models: Applications in Computer Vision, pp. 65–90. World Scientific (2001)
12. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)

13. Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., Qin, B.: Learning sentiment-specific word embedding for twitter sentiment classification. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), vol. 1, pp. 1555–1565 (2014)
14. Turian, J., Ratinov, L., Bengio, Y.: Word representations: a simple and general method for semi-supervised learning. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 384–394. Association for Computational Linguistics (2010)
15. Zhang, C., Ma, Y.: Ensemble Machine Learning: Methods and Applications, 1st edn. Springer, New York (2012). https://doi.org/10.1007/978-1-4419-9326-7