Journal of Computers Vol. 30 No. 2, 2019, pp. 183-191
doi:10.3966/199115992019043002017

An Automatic Case Review System Based on Deep Learning

Chen Li[1*], Zhenjiang Zhang[2], Bo Shen[2], Yajing Wang[3], Zhihong Ying[3], Yi-Chih Kao[4]

[1] School of Electronic and Information Engineering, Beijing Jiaotong University, 100044, Beijing, China
17120075@bjtu.edu.cn

[2] Key Laboratory of Communication and Information Systems, Beijing Municipal Commission of Education, Beijing 100044, China
{zhangzhenjiang, bshen}@bjtu.edu.cn

[3] Beijing Thunisoft Company, Beijing, China
{wangyajing, yingzhihong}@thunisoft.com

[4] Information Technology Service Center, National Chiao Tung University, Taiwan
ykao@mail.nctu.edu.tw

Received 6 December 2018; Revised 6 February 2019; Accepted 6 March 2019

Abstract. At present, the lawsuit documents provided by the parties during the court filing process still require to be reviewed through manual work. Faced with a large amount of lawsuit documents and limited reviewers, the review is less efficient. This paper proposes an automatic case review system based on deep learning, which can automatically review the lawsuit documents from the parties and identify the type of the case independently, instead of being manually reviewed. And it can recommend similar cases for reference by calculating the text similarity, which can help judges make fair decisions. In this paper, convolutional neural network (CNN), a deep learning model, is used to extract the key features for text classification in lawsuit documents and the features are input into the Softmax layer to classify the case. Then the system matches the case knowledge database to help the parties check for vacancies and provide suggestions, which realizes the automatic case review and improve the review efficiency.

Keywords: automatic case review, CNN, deep learning, recommendation of similar cases, text classification

1 Introduction

With the enhancement of people's legal awareness, more and more people choose to protect their legitimate rights and interests through litigation. In the traditional justice field, case review relies on the professional answers and debate processes of legally relevant individuals such as judges, lawyers, and prosecutors. For the public, the complicated legal provisions are obvious professional barriers. For professionals, most of the cases are common and simple cases, and the results are easier to handle, while a small number of them are complex cases, and the processing results are more difficult, resulting in a lot of manpower and material resources in actual cases. For legal practitioners, when dealing with cases, it is meaningful to recommend some historical cases which are very similar to the current cases. Finding historical cases related to current cases is one of the most important methods for legal practitioners to measure the judgement of defendants.

When the case is formally reviewed, it is mainly used to review the paper materials provided by the parties using manual review. At present, there are some expert systems in the judicial field. However, in the construction of the existing expert system, a large number of rules and help provided by legal

* Corresponding Author

professionals are still needed, and a lot of costs are also spent in the process of building the expert system. It is necessary to study an intelligent system which can effectively replace manual review to improve the efficiency of case review.

As the core of artificial intelligence, machine learning studies how to use machines to simulate human learning activities, including supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning. Its theory is mainly to design and analyze some algorithms that allow computers to automatically acquire rules from data, and use these rules to predict unknown data. Machine learning has been widely used in natural language processing, computer vision, data mining, information retrieval, speech recognition, medical diagnosis, robotics and other fields. Natural language processing is an important direction in the fields of machine learning. It studies various theories and methods that can realize effective communication between human and computer in natural language. Text categorization, as an important part of natural language processing, is an important basis for information retrieval and text mining. Its principle is to train the text of known categories by machine learning method, and construct a classification model, then use the obtained model to predict other text categories. The text categorization can help people organize text, manage text data and mine text information better by giving a categorization model and automatically categorizing text according to text content. It also improves greatly in accuracy and speed. Therefore, it has attracted more and more attention and has become one of the most important research directions in the field of information processing. There are many methods for text categorization, such as Naive Bayesian Classifier, K-Nearest Neighbor algorithm, Support Vector Machine (SVM) and Back Propagation Neural Network, which have achieved good results. In order to make the effect of text categorization better than the original, some scholars have made further improvements to the traditional text categorization algorithm. Generally speaking, these shallow network algorithms are faced with common limitations: local optimum, dimension disaster, over-fitting and so on, as well as the limited expression of complex functions in shallow networks with limited samples and computational units, which restricts their generalization ability in dealing with complex classification problems.

Text similarity is also a very important task in the research of NLP. There are many studies on text similarity. One is to design a series of features for text, then extract features, and conduct similarity research through the relationship between features. One is to map the text directly to the semantic vector. It maps the semantics of the text to the vector space by certain methods, and measures the distance or similarity of the original text at the semantic level by comparing their distance or similarity in space. In addition, topic models can also be used to calculate text similarity. By transforming text content into topic distribution, the similarity analysis of text content is carried out through similarity analysis of topic distribution.

With the development of artificial intelligence, Deep learning has attracted extensive attention from Internet data and artificial intelligence. Since Google, Microsoft, IBM, Baidu and other large Internet technology companies began to focus on the development of deep learning technology. By simulating the hierarchical structure of human brain, deep learning establishes a multi-layer neural network structure, extracts the distributed features of input data from the bottom to the top step by step, and finally establishes a good mapping function to describe the abstract relationship between the underlying signal and the high-level semantics. It has made great progress in the fields of computer vision [1], speech recognition [2], natural language processing [3] (NLP), data compression [4], target detection and tracking [5], information retrieval and machine translation. As a model in deep learning, convolutional neural network (CNN) can extract local features well. In addition, it greatly reduces the complexity of the network model and the number of training parameters, which greatly improves the training efficiency of Back Propagation [6] (BP) algorithm and shortens the training time. With the accumulation of time, a large number of cases in the judicial field are enough to meet the needs of large data for deep learning. On the other hand, with the continuous improvement of deep learning technology and the enrichment of models, many models and algorithms can effectively migrate to the issue of judicial intelligence.

In this paper, the CNN structure is used in the system to extract features from the lawsuit documents, and the Softmax layer is used to classify and obtain the case type. By calculating the extracted feature similarity, similar cases are recommended for judges. The system then matches the established case database with the type of case to help the parties improve the prosecution materials. For the judge, he can make a fair judgment with reference to the similar cases recommended.

The following is the structure of this paper. The second section is related work. The third section

introduces the automatic case review system. The fourth section is the experimental setup and results. The fifth section is the conclusion.

2 Related Work

In the earliest studies, researchers proposed a bag-of-words (BOW) model. The model maps the text into a vector. The value on each bit in the vector indicates the number of times the word appears in the text. By cooperating with machine learning methods such as support vector machine and naive Bayes, it achieves good results for text categorization tasks. In addition, it could calculate the TFIDF [7] of the word to get the weight, and then represent the text using the weight.

Curtotti et al. use machine learning to model the readability of legal texts by extracting context-free grammar and dependent syntactic information from legal texts as features, so as to improve the readability of law [8].

Kiritchenko et al. used n-grams information of words and characters to extract a large number of text features to classify the text [9]. N-gram is based on probability statistics, assuming that the nth word appears to be related to the first n-1 words and not to any other words. N-gram model splits the sequence of articles into groups through windows of size N. Then do statistics on these groups to form feature spaces and passed into classifiers for classification.

Word2vec is a Google open source tool for generating word vectors. It uses a two-layer neural network to map words to the same coordinate system and output them as numeric vectors [10].

Tang et al. used more abundant features in the research, such as Unigrams, Bigrams, text features, Average SG and other information [11]. Through the method of support vector machine, combined with these features to model, and achieved good results.

CNN can make better use of the local correlation of data to extract features, and have better performance in text classification. Kim proposed a text classification model, which just has one convolution layer and stitches word vectors into word matrices [12].

Zhang et al. used the character level convolution neural network method (CNN-char) to achieve remarkable results [13].

Kolawole et al. proposed a method of using rich features to calculate the text similarity [17]. In Kolawole's research, the lexical, syntactic and semantic features extracted from the text are synthesized. In lexical features, the matching information of strings is included first, and the differences at lexical level are determined by matching characters one by one in two texts. And Longest Common Substring, Levenshtein Distance and Jaccard Distance are also added into the model.

Liebeck et al. used word 2vec and LDA fusion method to study text semantic similarity [18].

In the judicial field, there have been some studies based on texts. Timmer et al. expressed and modeled complex probabilistic information through Bayesian belief network, and applied Bayesian reasoning of legal evidence through structure-oriented method [15].

Vlek et al. use Bayesian network to model and analyze the existing evidence of a case. In the trial of a criminal case, they can better extract the scene description of the current case from the text through the model, and can directly provide the judge or jury with the analysis of the existing evidence and the display of the results [14].

3 Model

In this section, the automated case review system will be introduced. Fig. 1 shows the overall structure of the system.

The text preprocessing module is mainly to remove stop words, which are some words that do not work for text analysis. Since the data we used is mainly in Chinese, we need to segment Chinese words. We know that in English, spaces are used as delimiters between words. Chinese inherits the tradition of ancient Chinese. Only characters, sentences and paragraphs can be simply delimited by clear demarcation marks. The single word does not have a formal demarcation mark. Some words are one character, and some words are composed of multiple characters. Although English also has the problem of phrase division, on the word level, Chinese is much more complicated and difficult than English. Chinese word

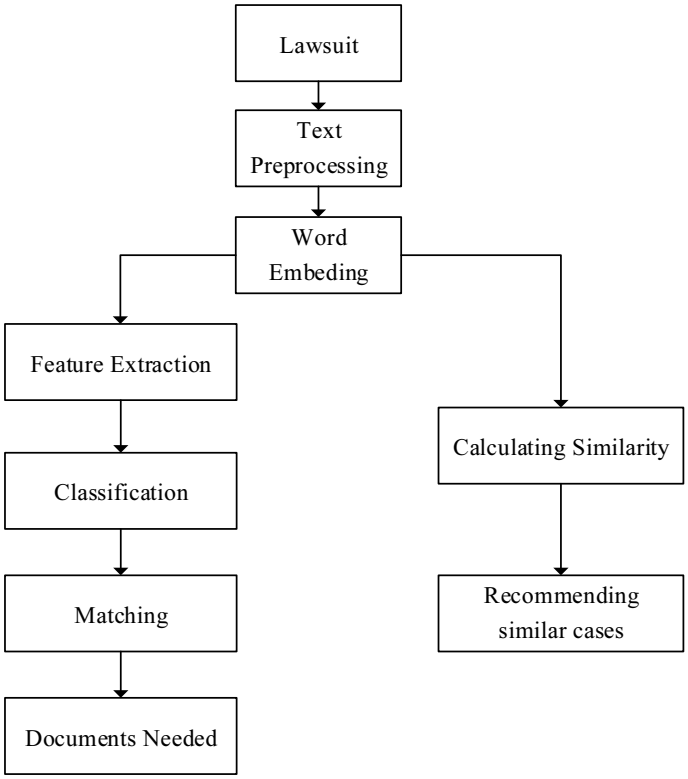

Fig. 1. The overall structure of the automated case review system

segmentation algorithms can be mainly divided into two categories: string matching method and statistics method.

String matching method. It matches the Chinese character strings to be analyzed with the entries in a sufficiently large dictionary according to certain strategies. If a string is found in the dictionary, the matching is successful.

Statistic method. Firstly, a large number of segmented texts are given, and statistical machine learning model is used to learn the rule of word segmentation so as to realize the segmentation of unknown texts.

This paper uses the jieba library, a valid Chinese word segmentation library which combines several algorithms to get a good segmentation result, to carry out Chinese word segmentation. Jieba's own dictionary library contains more than 20,000 words, most of which are the number of occurrences of the words that the authors have trained based on various experimental corporas such as the People's Daily, and the tagging of part of speech. The algorithms used by jieba word segmentation are as follows:

First, the trie tree structure is introduced when the word graph is scanned, and the directed wordless graph is used to analyze the condition of the word in the sentence. Generate a trie tree from the dictionary library. The generation process of the directed acyclic graph, in short, the dictionary search based on the self-contained dictionary library, and finally generates all the possibilities of segmenting the sentence. Second, the dynamic programming is introduced when the maximum probability path is searched, and the maximum segmentation combination based on word frequency is analyzed. For the words that are successfully segmented in the sentence to be segmented, the frequency of occurrence of the word is counted. The method of dynamically planning to find the maximum probability path is used for the maximum probability calculation of the sentence,

where the maximum probability of the sentence is reversed from right to left. Third, the introduction of Viterbi algorithm, combined with hidden Markov model to solve the problem of unregistered word recognition encountered in the process of Chinese word segmentation. The BEMS of Chinese words respectively represent the start state, the end state, the middle position mark, and the singgle's expression without any context, and can be singularly. Three probability tables are available based on continuous training of large corpora.

The word embedding module converts words into word vectors. In this paper, we use the word2vec model to get the vectors. In order to solve the problem of word vector sparsity and reduce the dimension of word vector, word2vec uses a two-layer neural network. We use Continuous Bag-of-Words (CBOW)

model to train the neural network. The input is the word vectors corresponding to the context-related words, and the output is the word vector we want. Fig. 2 shows the structure of the system.

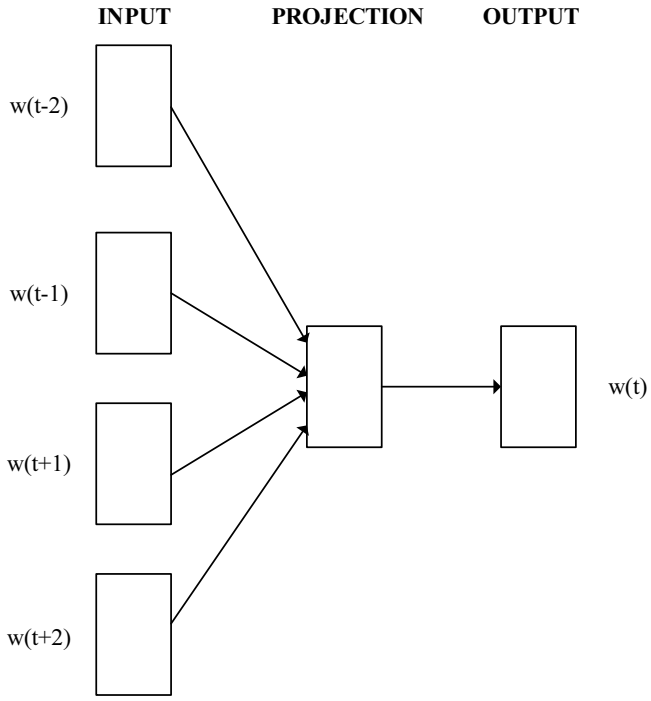

Fig. 2. The structure of the CBOW

The feature extraction and classification module, which is the CNN structure, is shown in Fig. 3. The input is composed of word matrix denoted as I. Let i_n be the n-th word vector in I. The dimension of each word is d dimensional. Since the length of the text information is not the same, n is not fixed.

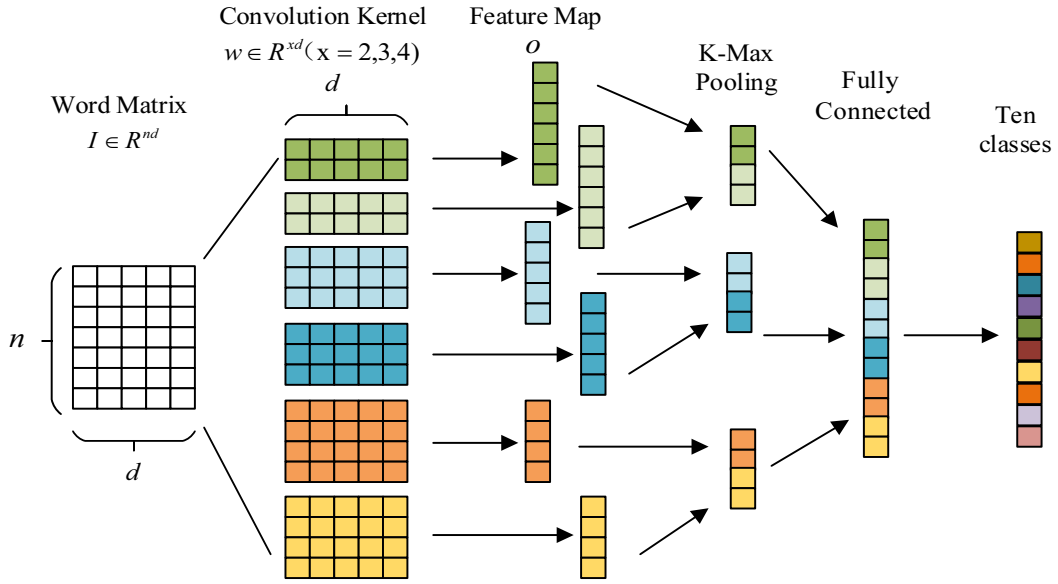

Fig. 3. The CNN structure

In convolution layer, the convolution kernel has the same dimensions as the word in order to guarantee that the model takes the word as the granularity which could extract richer features. Let $w \in R^{xd}$ be the convolution kernel weight. And we use convolution kernels of different sizes and quantities to extract multiple features. The n-th feature o_n is generated from a window of words $i_{n:n+x-1}$ by:

$$o_n = f(w \cdot i_{n:n+x-1} + b) \quad \textbf{(1)}$$

Here b is a bias and f is a non-linear function and the stride is 1. After the convolution operation applied to the whole word matrix, we get the $(n-x+1)$dimensional vector, that is, the feature map:

$$o = \{o_1, o_2, \ldots, o_{n-x+1}\} \quad \textbf{(2)}$$

We then use a k-max pooling [16] operation over the feature map which is to select top-k features in the feature map and keep these values in order. This is to preserve important and strong features and positional information. After pooling, the output o is the top k values of o.

Then the features are passed to a fully connected layer with dropout and Softmax output. Some nodes in the neural network are discarded randomly with a certain probability, and the discarding is temporary. Because it is randomly discarded, the node may continue to work on the next input sample. By using this method, the over-fitting of the model will be greatly reduced.

After classification, according to the types of cases, the system can query the documents needed by the type from the case knowledge database to help the parties check for vacancies and provide suggestions.

In order to recommend similar cases, it is necessary to map the case description text to vector space, and then calculate the similarity between the vectors through a certain measure, so as to rank and recommend.

We have obtained word embedding information of words through word2vec, which realizes the representation from words to vectors. Another thing to do is to complete the mapping of text to vector. In this paper, word embedding and TF-IDF are weighted averaged.

TF-IDF is a commonly used statistical method in text mining and information retrieval. TF is word frequency, which refers to the frequency of a specific word appearing in a given document. The calculation method is shown in formula 3. Where $n_{i,j}$ denotes the total number of occurrences of the word t_i in the jth document and $tf_{i,j}$ denotes the frequency of the word t_i in the jth document.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \quad \textbf{(3)}$$

IDF is the frequency of reverse documents, which measures the popularity of words in various documents. The calculation method is shown in formula 4. Where $|D|$ is the total number of documents and denominator is the number of documents containing the word t_i. Since the denominator equals zero if the word t_i does not appear in the document, the denominator in formula 4 is usually added to 1.

$$idf_i = \log \frac{|D|}{|\{j : t_i \in d_j\}|} \quad \textbf{(4)}$$

Therefore, TF-IDF is calculated as a product of the word frequency of each word and the word frequency of the reverse document, as shown in formula 5.

$$tfidf_{i,j} = tf_{i,j} * idf_i \quad \textbf{(5)}$$

In this paper, each word in a text has its own embedding vector and TF-IDF value. The embedding of a text is the average of the product of embedding and TF-IDF of each word. Thus, the influence of some common words on text embedding is neglected. As shown in formula 6, d_j is the embedding representation of the case in chapter j. emb_i is the word embedding of the i-th word in the case of chapter j. $tfidf_{i,j}$ denotes the TF-IDF value of the i-th word in the case of chapter j.

$$d_j = \frac{1}{N} \sum_{i=1}^{N} emb_i * tfidf_{i,j} \quad \textbf{(6)}$$

In order to calculate the similarity, the appropriate distance measurement method is very important. Common distance measurement methods include euclidean distance, cosine similarity and so on.

Euclidean Distance is the most common measure. The calculation method is shown in formula 7. When using Euclidean distance to measure similarity, it needs to be normalized between 0 and 1, so it

needs some processing, so it is usually shown in formula 8.

$$D(X,Y) = \sqrt{\sum_{i=1}^{N}(x_i - y_i)^2} \tag{7}$$

$$similarity = \frac{1}{1 + D(X,Y)} \tag{8}$$

Cosine Similarity is a cosine value that calculates the angle between vectors to measure the distance and similarity between two vectors. The calculation method is shown in formula 9. Since $\cos\theta$ ranges from - 1 to 1, it is more desirable to normalize it to 0 to 1 in practical use. Therefore, when calculating cosine similarity, the transformation formula is usually shown in formula 10

$$\cos\theta = \frac{X \cdot Y}{\|X\|\|Y\|} \tag{9}$$

$$similarity = 0.5 + 0.5 * \cos\theta = 0.5 + 0.5 * \frac{X \cdot Y}{\|X\|\|Y\|} \tag{10}$$

Cosine similarity is widely used in natural language processing, especially in computing the similarity between two documents, because it is not related to the length of the vector, but only to the angle between the vectors.

4 Experimental Setup and Results

4.1 Dataset

We collected about 10000 lawsuits for a total of 10 categories, including marriage, traffic accident, contract dispute, criminal defense, debt, industrial injury, relocation, medical dispute, consumption rights and interests and land dispute. We finally got about 30000 lawsuits by using data enhancement through synonym replacement. We randomly selected 10% of the data as a test set.

4.2 Word Vectors

After removing the stop words from the text and after the Chinese word segmentation, we get a corpus of words. Here we use the word2vec model, using the CBOW training method [8], which infers the word according to the context. Then we get the 64-dimensional word vectors.

4.3 Experimental Setup and Results

We choose the rectified linear units as the non-linear function. The size of convolution kernel is set as 2,3,4,5 with 128 feature maps each. The dropout rate is set as 0.5. The mini-batch size is set as 64. The total number of categories is 10. The embedding dimension is 64.

The result of the model are listed in Table 1. From the table we can see that the average recall is 0.875 and the average precision is 0.873 which can realize the automatic classification of the lawsuits text to a certain extent. However, due to insufficient data, the effect of classification needs to be improved.

In the research centers recommended by similar cases, three methods are tried, namely, tf-idf, word2vec direct averaging and word 2vec fusion of tf-idf. The three methods are to map the case text to vector space. The word embedding method which fuses TF-IDF is to weigh the words of TF-IDF as coefficients. In this paper, Discounted cumulative gain (DCG) was selected as an indicator of the evaluation model. Twenty test cases were randomly selected for DCG@5 evaluation. For each query case, the value of DCG@5 can be given, so avg-DCG@5 is chosen as the evaluation index of the model. The results are shown in Table 2.

Table 1. The recall rate and the accuracy

	Recall	Precision
Debt	0.87	0.89
Marriage	0.89	0.89
Criminal	0.91	0.90
Relocation	0.86	0.86
Land Dispute	0.87	0.88
Traffic Accident	0.90	0.89
Industrial Injury	0.87	0.82
Medical Dispute	0.89	0.87
Contract Dispute	0.84	0.85
Consumption Rights	0.85	0.88
Average	0.875	0.873

Table 2. Similar case recommendation experiment results

The model	Precision
tf-idf	14.73
word2vec averaging	12.49
word2vec+tf-ifd	17.81

From the results in Table 4-2, we can see that in similar case recommendation tasks, the doc2vec method still has a gap with the traditional TF-IDF method on avg-DCG@5, while the word2vec method (wor2vec+tf-idf) which fuses TF-IDF has the best effect.

5 Conclusion

In this paper, we propose an automatic case review system. We use the CNN to extract the lawsuit texts features and classify the case based on them and got a better classification result. After matching the case knowledge database by the type of the case, the system could help the parties check for vacancies and provide suggestions, realizing the automatic case review. We also made similar recommendations for the case. By comparing several different models, we found that using word2vec+tf-ifd as the article vector can get better recommendations.

In the future, on the one hand, we will collect more data to train a model with higher classification accuracy, and we may try to classify the text by character rather than word and a deeper model. On the other hand, in researching the recommendation of similar cases, we will integrate more models to achieve better results.

Acknowledgements

This research was funded by the National Key Research and Development Program of China, grant number 2018YFC0831300, and the Fundamental Research Funds for the Central Universities, grant number 2017JBZ107.

References

[1] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Proc. Advances in neural information processing systems, 2012.

[2] A. Graves, A. Mohamed, G.E. Hinton, Speech recognition with deep recurrent neural networks, in: Proc. 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, 2013.

[3] Y. Bengio, R. Ducharme, P. Vincent, C. Jauvin, A neural probabilistic language model, Journal of machine learning research 3(Feb.)(2003) 1137-1155.

[4] S. Han, M. Huizi, W.J. Dally, Deep compression: compressing deep neural networks with pruning, trained quantization and huffman coding <http://arxiv.org/abs/1510.00149>, 2015.

[5] R. Girshick, D. Jeff, D. Trevor, M. Jitendra, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proc. the IEEE Conference on Computer Vision and Pattern Recognition, 2014.

[6] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors, Cognitive Modeling 5(3)(1988) 1.

[7] B. Trstenjak, S. Mikac, D. Donko, KNN with TF-IDF based framework for text categorization, Procedia Engineering 69(2014) 1356-1364.

[8] M. Curtotti, E. McCreath, T. Bruce, Machine learning for readability of legislative sentences, in: Proc. the 15th International Conference on Artificial Intelligence and Law, 2015.

[9] S. Kiritchenko, X. Zhu, S.M. Mohammad, Sentiment analysis of short informal texts, Journal of Artificial Intelligence Research 50(2014) 723-762.

[10] T. Mikolov, I. Sutskever, K. Chen, Distributed representations of words and phrases and their compositionality, in: Proc. Advances in Neural Information Processing Systems, 2013.

[11] D. Tang, B. Qin, T. Liu, Document modeling with gated recurrent neural network for sentiment classification, in: Proc. the 2015 Conference on Empirical Methods in Natural Language Processing, 2015.

[12] Y. Kim, Convolutional neural networks for sentence classification. <http://arxiv.org/abs/1408.5882>, 2014.

[13] X. Zhang, J. Zhao, Y. LeCun, Character-level convolutional networks for text classification, in: Proc. Advances in Neural Information Processing Systems, 2015.

[14] C. Vlek, H. Prakken, S. Renooij, Constructing and understanding bayesian networks for legal evidence with scenario schemes, in: Proc. the 15th International Conference on Artificial Intelligence and Law, 2015.

[15] S.T. Timmer, J.J.C. Meyer, H. Prakken, A structure-guided approach to capturing bayesian reasoning about legal evidence in argumentation, in: Proc. the 15th International Conference on Artificial Intelligence and Law, 2015.

[16] N. Kalchbrenner, E. Grefenstette, P. Blunsom, A convolutional neural network for modelling sentences. <http://arxiv.org/abs/1404.2188>, 2014.

[17] L. Di Caro, G. Boella, Normas at semeval-2016 task 1: SEMSIM: a multi-feature approach to semantic text similarity, in: Proc. the 10th International Workshop on Semantic Evaluation (SemEval-2016), 2016.

[18] M. Liebeck, P. Pollack, P. Modaresi, Hhu at semeval-2016 task 1: multiple approaches to measuring semantic textual similarity, in: Proc. the 10th International Workshop on Semantic Evaluation (SemEval-2016), 2016.