Legal Case Inspection: An Analogy-Based Approach to Judgment Evaluation

Shang Li(✉), Bin Guo, Yilei Cai, Lin Ye, Hongli Zhang, and Binxing Fang

School of Computer Science and Technology,
Harbin Institute of Technology, Harbin 150001, China
ls@hit.edu.cn

Abstract. In the era of big data, enormous growth of various legal data leads to a huge burden on law professionals, which lies in the contradiction between the increasing number of legal cases and the shortage of judicial resources. This issue enlightens us to explore the key technologies in the computer-aided criminal case process lines. In this paper, we investigate an analogy-based method of legal case inspection. We use the document vector generated by Doc2Vec (semantics-based case feature, SCF) and the feature defined by the case judgement model (model-based case feature, MCF) as two ways to find similar cases. The measurement methods of similarity between two cases and the deviation of case judgment are also defined. Experimental results on a real-world dataset shows the effectiveness of our method. The recall rate of irrational cases when using the MCF is higher than that when using the SCF.

Keywords: Legal case inspection · Judgment evaluation · Analogy · Neural networks

1 Introduction

Legal case inspection (LCI) is an inspection mechanism for the trial work of the people's courts and judges at all levels. As the number of legal cases continues to grow in recent years, miscarriage of justice or judicial arbitrariness may come up in judicial practice, resulting from inconsistent judgment scale of different judges due to their different understandings and interpretation of the laws and cases, the mistakes of the judges at work, etc. The LCI aims to find irrational judicial decisions in order to avoid trial bias as far as possible. If the automated LCI process can be implemented, it can effectively prevent miscarriage of justice or judicial arbitrariness in the future, which is of great significance to the maintenance of judicial justice.

In this paper, we investigate the analogy-based LCI method, which is to evaluate the judgment rationality of the target case via the judgment results of its precedent similar cases. Formally, the input of the LCI method is the case text excluding the result of judgment, while the output is the deviation of prison term. The deviation needs to be measured by similar cases, while similar cases need to be determined by the similarity of the case features. Therefore, the LCI process can be divided into three steps: case feature generation, similarity measurement, and deviation measurement. Experimental

X. Sun et al. (Eds.): ICAIS 2019, LNCS 11632, pp. 148–158, 2019.
https://doi.org/10.1007/978-3-030-24274-9_13

results on a real-world dataset containing more than 40,000 judgment documents of theft cases shows the effectiveness of our method.

The rest of this paper is organized as follows. Section 2 overviews related work. The generation of two types of case feature is described in Sect. 3. In Sect. 4, we propose the LCI method. Evaluation results are presented in Sect. 5, and finally, Sect. 6 contains the concluding remarks.

2 Related Work

Several key issues have been studied in the field of artificial intelligence and law, such as the prediction of judgment result and the inference of relevant law articles.

The judgment prediction is an important combination point between artificial intelligence and law. Liu et al. [1, 2] use the KNN algorithm to classify 12 criminal cases in Taiwan, the former establishes a case-based reasoning system by defining various criminal rules in advance, while the latter implement a classifier that uses phrases to index. To obtain a better classification effect, Sulea et al. [3, 4] increased the number of classifiers, they combine with the output of multiple SVM (support vector machine) classifiers to predict the law area, ruling and sentence. Katz et al. [5] propose a model with the extreme random tree to predict the voting behavior of the US Supreme Court from 1953 to 2013. Lin et al. [6] achieve a deeper understanding of the factors of the case by adding some non-TF-IDF features. According to the three characteristics of relevant legal provisions, sentiment analysis of crime facts and prison term, Liu and Chen [7] use SVM algorithm to classify the judgment text automatically.

Inferring applicable law articles for a given case is also an important work. Aletras et al. [8] propose a binary classification model, and the target output is a practical judgment on whether there is a violation of a specific clause of the Human Rights Convention. Liu and Liao [9] convert this multi-label problem into a multi-category classification problem by considering a fixed set of legal provisions, and achieve satisfactory initial results in the classification of theft and gambling crimes. Liu et al. [10] propose a more optimized algorithm, they first use the SVM for preliminary articles classification, and then use the word-level features and co-occurrence tendencies in articles to reorder the results.

In summary, previous studies have considerably facilitated the advance in the field of artificial intelligence and law [11–18]. Nevertheless, legal case inspection remains a huge challenge. Our work in this paper aims to fill this gap.

3 Feature Generation

The LCI process is based on a set of similar cases of the target case, and the similarity measurement calls for calculable case features. In this section, we present the generation method of two types of case feature, i.e. the semantics-based case feature (SCF) and the model-based case feature (MCF).

3.1 Semantics-Based Case Feature

Considering that the inspection is an unsupervised process, the case feature used for inspection should be obtained without relying on the process of inspection. The training process of the full text through Doc2Vec is an unsupervised learning process, and the document vector is a one-dimensional vector, which can be directly calculated, so the document vectors are more suitable to construct the case feature.

For the SCF, the training process based on Doc2Vec is shown in Fig. 1. Firstly, each case is transformed into a long sentence. Then, texts of all the cases is used as a corpus and input into Doc2Vec for training. After training, the document vectors corresponding to each case can be obtained.

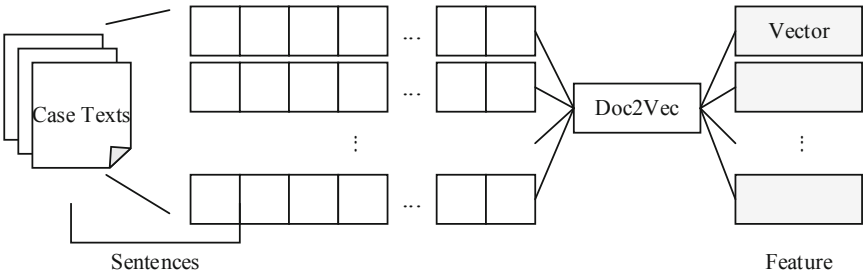

Fig. 1. Generation of SCF

3.2 Model-Based Case Feature

Taking the theft case as the research object in this paper, we need first build its judgment model according to relevant law articles, i.e. Article 264 in the Criminal Law of the People's Republic of China. Through the comprehensive analysis of Article 264 and the structure of judgment document, we can describe a theft case with 11 dimensions as the judgment model shown in formula (1): (a) the value of stolen items, (j) whether the defendant is juvenile, (d) whether the defendant is disabled, (b) whether the crime can be deemed as burglary (breaking in home), (w) whether the defendant carried lethal weapons, (p) whether the defendant is a pickpocket, (o) whether the crime involves other serious circumstances (including but not limited to: collision, arson, resistance to arrest, etc.), (r) whether the defendant is a recidivist, (c) whether the defendant returned stolen items or compensated the victim, (s) whether the defendant voluntarily surrendered and (t) the prison term.

$$C = (a, j, d, b, w, p, o, r, c, s, t) \tag{1}$$

Figure 2 briefly illustrates the generation process of the MCF based on GRU (Gated Recurrent Unit) network. When case text arrives, the feature generator divides it into sentences and deals with the sentences one by one. Through the Chinese word segmentation and word embedding step, words will be converted to distributed representations as the k-dimensional vector form. Then the GRU network takes the

sequence of word vectors as input, and generates an output sequence of vectors through GRU units. Finally, the feature merging part averages the features of all the sentences to transform them into a single vector as the MCF.

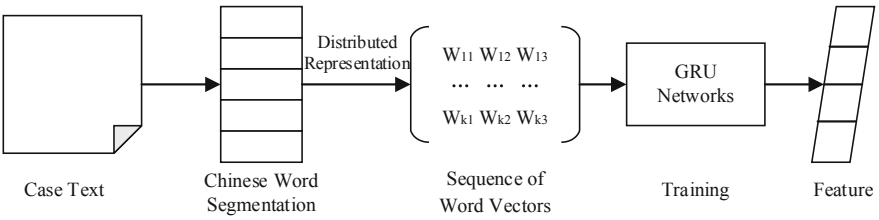

Fig. 2. Generation of MCF

4 Method

4.1 Similarity Measurement

Before performing LCI process, it is necessary to obtain similar cases of the target case. The key issue is how to measure the similarity between two judgment documents. Generally, vector-based similarity can be computed by one of these metrics: Euclidean distance, Manhattan distance, Minkowski distance, cosine similarity, and Jacquard similarity coefficient.

The Euclidean distance is the linear distance between two points in the Euclidean space and can be calculated as follows:

$$D(X,Y) = \sqrt{\sum_{i=1}^{n} (X_i - Y_i)^2} \tag{2}$$

The Manhattan distance is the sum of the absolute value of the difference between each dimension of the vectors as follows:

$$D(X,Y) = \sum_{i=1}^{n} |X_i - Y_i| \tag{3}$$

The Minkowski distance is a metric in a normed vector space which can be considered as a generalization of both the Euclidean distance and the Manhattan distance, which can be calculated as follows:

$$D(X,Y) = \left(\sum_{i=1}^{n} |X_i - Y_i|^k \right)^{\frac{1}{k}} \tag{4}$$

The cosine similarity is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them as follows:

$$sim(X,Y) = \frac{X \cdot Y}{\|X\|\|Y\|} = \frac{\sum_{i=1}^{n}(X_i Y_i)}{\sqrt{\sum_{i=1}^{n} X_i^2}\sqrt{\sum_{i=1}^{n} Y_i^2}} \tag{5}$$

The Jaccard similarity coefficient can measure similarity between two sets, X and Y, each with n boolean attributes. Each attribute of X and Y can either be 0 or 1. The Jaccard similarity coefficient is calculated as follows:

$$J(X,Y) = \frac{M_{11}}{M_{01} + M_{10} + M_{11}} \tag{6}$$

where M_{11} is the total number of attributes where X and Y both have a value of 1, M_{01} and M_{10} represent the total number of attributes where the attribute of X is 0/1 and the attribute of Y is 1/0, respectively.

According to the definitions of Euclidean distance, Manhattan distance, and Minkowski distance, they focus on the absolute difference of each dimension in the vector. For the case features, since different dimensions represent different information, it is more suitable to measure the similarity by relative difference. Therefore, we select the cosine similarity to calculate the similarity between cases with SCF. As for cases with MCF, considering that the nine features except the value of stolen items are boolean values, the strategy is to multiply the similarity measured by the 9 boolean features and the value of stolen items, respectively. The similarity measured by the 9 boolean features is calculated as Eq. (7), that is, the number of attributes with the same boolean value is divided by the total number of attributes.

$$sim(X,Y) = \frac{|\{i|X_i = Y_i, i = 0,1,2\cdots n\}|}{n} \tag{7}$$

For SCF, it is necessary to multiply the similarity of the document vector by the similarity of the value of stolen items to obtain the final similarity. For MCF, it is necessary to multiply the similarity of boolean features by the similarity of the value of stolen items. Considering that the relative difference of theft value can better reflect the similarity degree than the absolute difference, the relative difference is selected to measure the similarity. Since the value of stolen items is mostly concentrated below 10,000 yuan, in order to have its value evenly distributed, we take its logarithm to when calculating the similarity. Therefore, the similarity of the value of stolen items is calculated as follows:

$$sim(x,y) = 1 - \frac{|\log(x) - \log(y)|}{\max(\log(x), \log(y))} \tag{8}$$

In summary, the similarity between cases with SCF and MCF is calculated as Eqs. (9) and (10), respectively.

$$sim(X, Y) = sim_{docvec}(X_{docvec}, Y_{docvec}) \cdot sim_{money}\left(X_{money}, Y_{money}\right) \tag{9}$$

$$sim(X, Y) = sim_{boolean}(X_{boolean}, Y_{boolean}) \cdot sim_{money}\left(X_{money}, Y_{money}\right) \tag{10}$$

4.2 Deviation Measurement

In the LCI process, in order to evaluate whether the judgment result is rational or not and to compare the difference in the judgment rationality between different cases, the judgment deviation of a legal case should be defined to measure the likelihood of a certain case's misjudgment. The judgment deviation is calculated based on a set of similar cases to the target case, which are generated according to the similarity measurement. We hold the point that the judgment deviation of a case is closely related to judgment results of its similar cases, that is, if the judgment difference between a case and others exceeds a certain degree, the case's judgment may be irrational. The judgment deviation of the target case t is calculated as Eq. (11): firstly, the top K similar cases to t are selected, then the relative errors between t and each one of K cases are calculated, respectively, and finally, the average value of the relative errors is taken as the judgment deviation of the case t.

$$E_t = \frac{\sum_{i=1}^{K} \frac{|p_i - p_t|}{p_t}}{K} \tag{11}$$

5 Experiments

In order to verify the effectiveness of the proposed LCI method, the SCF and the MCF are used in the experimental evaluation.

5.1 Evaluation of Semantics-Based Case Feature

In the evaluation, Dov2Vec is used to train the case text to construct features. The corpus used for training consists of 41,418 judgment documents, and the document vectors of 50, 100, 150, 200, 250 and 300 are trained, respectively. For each case, the similarities to the other cases are calculated according to the method presented in Sect. 4.1 with the six types of document vectors. After the similarity measurement, all the cases are sorted in descending order of the similarity to the target case.

Similarity has the following effect on the judgment deviation: the higher the similarity to the target case, the more realistic the deviation obtained, and vice versa. In order to obtain a reliable judgment deviation, it is necessary to select an appropriate document vector dimension to maximize the credibility of similarity measurement. Here, we employ the average relative error (ARE) of the judgment results of each case and its most similar case as the metric to compare the effects of different dimensional document vectors on similarity measurement. As is shown in Fig. 3, the ARE achieves

the minimum value at 100 dimensions, indicating that the similarity calculated with the 100-dimensional document vector is more reliable than other dimensions, so the 100-dimensional document vector is selected as the SCF.

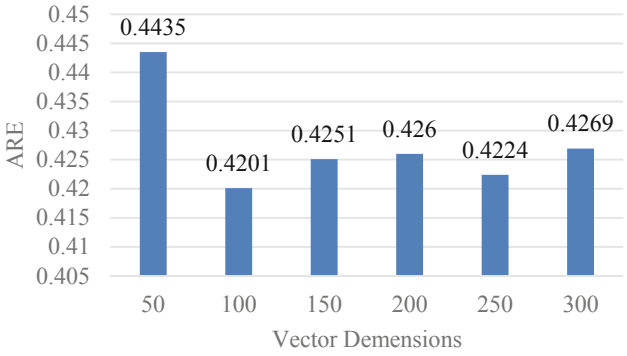

Fig. 3. Comparison of ARE on different vector dimensions

To obtain the judgment deviation of each case, we first select the top K similar cases to the target case (K = 10, 20, …, 100), then calculate the relative error between each case and the target case, respectively, and finally, use the ARE as the judgment deviation of the target case. Here, we employ the number of cases with judgment deviation greater than 1.0 as the metric to evaluate the effects of different Ks on the effectiveness of LCI method. As is shown in Fig. 4, the number of cases with judgment deviation greater than 1.0 achieves the maximum value of 148 when K = 10.

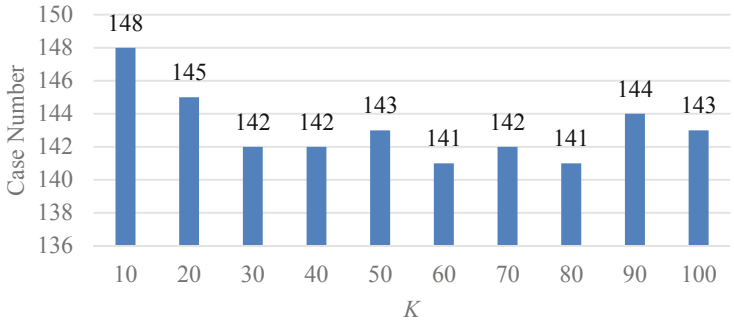

Fig. 4. Number of cases with deviation greater than 1.0 with different Ks

In order to validate whether the cases selected based on the judgement deviation are actually irrational, we manually read the cases with judgment deviation greater than 1.0 and make reference to relevant law articles. When K varies from 10 to 100, we get a total of 189 cases with judgment deviation greater than 1.0, 125 of which considered as actually irrational after manual validation. According to the number of actually

irrational cases found with ten *K*s, we calculate the recall rate, precision and the corresponding F1 score, respectively.

As is shown in Table 1, the highest recall rate is obtained when $K = 10$ or $K = 20$, the highest precision is obtained when $K = 50$, and the highest F1 value is obtained when $K = 20$. Considering that the inspection is more focused on finding irrational cases, the highest recall rate should be obtained as far as precision is acceptable. On the whole, when $K = 20$, the recall rate is the highest, and the precision is also higher. Therefore, it is best when $K = 20$, that is, supervise through 20 cases with the highest similarity to the target case.

Table 1. Comparison of recall rate, precision and F1 score with different *K*s

K	Recall	Precision	F1
10	**0.8400**	0.7094	0.7692
20	**0.8400**	0.7241	**0.7777**
30	0.8080	0.7112	0.7565
40	0.8080	0.7112	0.7565
50	0.8320	**0.7272**	0.7761
60	0.8000	0.7092	0.7518
70	0.8000	0.7042	0.7490
80	0.7920	0.7021	0.7443
90	0.8240	0.7152	0.7657
100	0.8160	0.7132	0.7611

5.2 Evaluation of Model-Based Case Feature

The experimental setting of MCF evaluation is the same as the SCF evaluation. Let $K = 10, 20, \ldots, 100$, the numbers of cases with judgment deviation greater than 1.0 are shown in Fig. 5. The results show that the number of cases with judgment deviation greater than 1.0 is 156 at $K = 10$, and as the value of K increases, the number of cases gradually decreases, and finally, the minimum number of cases is 121 at $K = 100$.

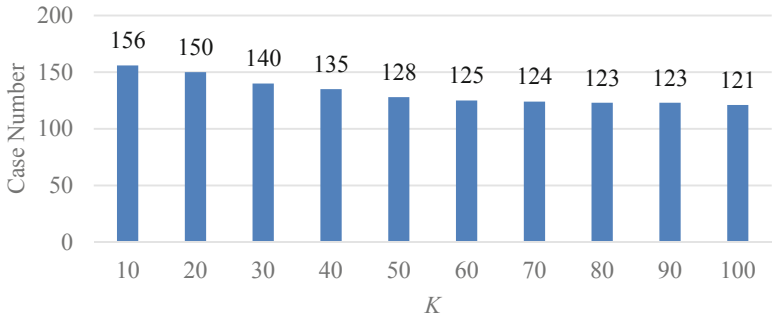

Fig. 5. Number of cases with deviation greater than 1.0 with different *K*s

A total of 200 cases are collected from 10 sets of cases. After manual validation, 137 cases are considered to be actually irrational, and the other 63 cases were rational. As is shown in Table 2, when $K = 10$ or $K = 20$, the highest recall rate is 0.8029, and when $K = 90$, the highest precision is 0.7886, and when $K = 30$, the highest F1 score is 0.7870. Although the precision is lower at $K = 20$, the recall rate achieves the maximum value. In the judicial practice, the LCI task requires a higher recall rate, so the best effect can be achieved when $K = 20$.

Table 2. Performance with different *K*s

K	Recall	Precision	F1
10	**0.8029**	0.7051	0.7508
20	**0.8029**	0.7333	0.7665
30	0.7956	0.7785	**0.7870**
40	0.7664	0.7777	0.7720
50	0.7226	0.7734	0.7471
60	0.7080	0.7760	0.7404
70	0.7080	0.7822	0.7432
80	0.7007	0.7804	0.7384
90	0.7080	**0.7886**	0.7461
100	0.6934	0.7851	0.7364

5.3 Case Study and Analysis

When using the SCF, the recall rate reaches the highest value of 0.8400 when the top 20 similar cases are used for LCI. When using the MCF, the recall rate reaches the highest value of 0.8029 when the top 20 similar cases are used for LCI. Intuitively, the effect of the former is better. However, due to the different experimental sets used to calculate the recall rate, the comparison results cannot correctly reflect the actual differences between the two features.

There are 254 cases with judgment deviation greater than 1.0, of which 157 irrational cases and 97 rational cases were found via manual validation. The 254 cases can be used as the dataset to compare the recall rate of the two features. As is shown in Fig. 6, when using the MCF and $K = 10$ or $K = 20$, there is the highest recall rate, whose value is 0.7006. It can be seen that when using the MCF for LCI, the recall rate is higher than the SCF, so the feature is more suitable for LCI.

Considering that the case with high judgment deviation is an irrational case that does not meet the judgment rules of most cases in the judicial system, but because the service object of the inspection technology is the judicial supervisor, it is still necessary for the supervisor to analyze and evaluate whether the judgment result is actually rational, rather than the system to judge whether it is rational. What the system can do is to automatically screen out the abnormal cases and submit them to the supervisors for review, so as to avoid supervisors looking at all the cases, and improve the work efficiency of supervisors. From this point of view, if the system can screen out the cases with problems as much as possible and submit them to the supervisors, the effect of

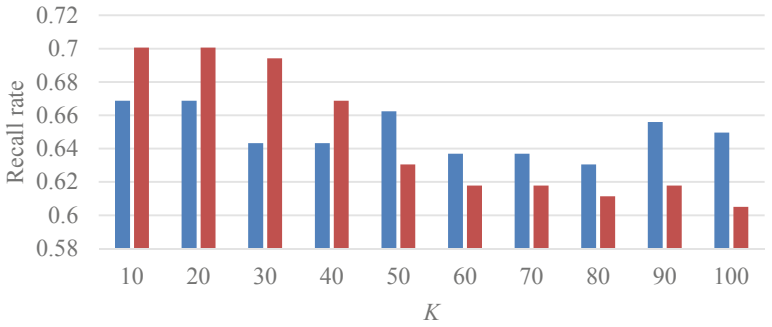

Fig. 6. Comparison of recall rates with two case features

automated inspection will be achieved. The experiments in this section show that under the optimal conditions, the inspection method defined in this paper can screen out 110 cases from 157 cases with problems, and realize the expected function to a certain extent, and initially achieve the automatic inspection.

6 Conclusion

In this paper, we investigated an analogy-based approach to legal case inspection. The SCF is trained via Doc2Vec, and the MCF is generated according to the judgment model of theft cases. Then, the similarity calculation methods of the two features are respectively defined, and the final similarity is the result of multiplying the feature similarity by the stolen goods value similarity. In order to quantify the deviation of the case, its calculation method is defined. The experimental results show that no matter which feature is used, the best results are obtained when using the top 20 similar cases of the target case, and the results of the two characteristics on the same dataset show that the recall rate based on the case model is the better, which is considered to be more suitable for LCI task. In future work, we will further validate and advance the proposed method on various types of criminal case.

Acknowledgment. This work is supported by the National Key Research and Development Program of China under grants 2018YFC0830902 and 2016QY03D0501, and the National Natural Science Foundation of China (NSFC) under grants 61723022 and 61601146.

References

1. Liu, C., Chang, C., Ho, J.: Case instance generation and refinement for case-based criminal summary judgments in Chinese. J. Inf. Sci. Eng. **20**(4), 783–800 (2004)
2. Liu, C.-L., Hsieh, C.-D.: Exploring phrase-based classification of judicial documents for criminal charges in Chinese. In: Esposito, F., Raś, Z.W., Malerba, D., Semeraro, G. (eds.) ISMIS 2006. LNCS (LNAI), vol. 4203, pp. 681–690. Springer, Heidelberg (2006). https://doi.org/10.1007/11875604_75

3. Sulea, O., Zampieri, M., Malmasi, S., Vela, M., Dinu, L.P., van Genabith, J.: Exploring the use of text classification in the legal domain. In: Proceedings of the Second Workshop on Automated Semantic Analysis of Information in Legal Texts (2017)
4. Sulea, O., Zampieri, M., Vela, M., van Genabith, J.: Predicting the law area and decisions of French Supreme Court cases. In: Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2017), pp. 716–722 (2017)
5. Katz, D.M., Bommarito II, M.J., Blackman, J.: A general approach for predicting the behavior of the supreme court of the United States. PLoS ONE **12**(4), e0174698 (2017)
6. Lin, W., Kuo, T., Chang, T., Yen, C., Chen, C., Lin, S.: Exploiting machine learning models for Chinese legal documents labeling, case classification, and sentencing prediction. Comput. Linguist. Chin. Lang. Process. **17**(4), 49–68 (2012)
7. Liu, Y., Chen, Y.: A two-phase sentiment analysis approach for judgement prediction. J. Inf. Sci. **44**(5), 594–607 (2018)
8. Aletras, N., Tsarapatsanis, D., Preotiuc-Pietro, D., Lampos, V.: Predicting judicial decisions of the European Court of Human Rights: a natural language processing perspective. PeerJ Comput. Sci. **2**, e93 (2016)
9. Liu, C.-L., Liao, T.-M.: Classifying criminal charges in Chinese for web-based legal services. In: Zhang, Y., Tanaka, K., Yu, J.X., Wang, S., Li, M. (eds.) APWeb 2005. LNCS, vol. 3399, pp. 64–75. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-31849-1_8
10. Liu, Y., Chen, Y., Ho, W.: Predicting associated statutes for legal problems. Inf. Process. Manage. **51**(1), 194–211 (2015)
11. Meng, R., Rice, S., Wang, J., Sun, X.: A fusion steganographic algorithm based on faster R-CNN. CMC Comput. Mater. Continua **55**(1), 1–16 (2018)
12. Wang, R., Shen, M., Li, Y., Gomes, S.: Multi-task joint sparse representation classification based on fisher discrimination dictionary learning. CMC Comput. Mater. Continua **57**(1), 25–48 (2018)
13. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
14. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014), pp. 1724–1734 (2014)
15. Palau, R.M., Moens, M.: Argumentation mining: the detection, classification and structure of arguments in text. In: Proceedings of the 12th International Conference on Artificial Intelligence and Law, pp. 98–107 (2009)
16. Hachey, B., Grover, C.: Extractive summarization of legal texts. Artif. Intell. Law **14**(4), 305–345 (2006)
17. Farzindar, A., Lapalme, G.: Legal text summarization by exploration of the thematic structures and argumentative roles. In: Proceedings of the Text Summarization Branches Out Workshop, pp. 27–38 (2004)
18. Galgani, F., Compton, P., Hoffmann, A.: Combining different summarization techniques for legal text. In: Proceedings of the Workshop on Innovative Hybrid Approaches to the Processing of Textual Data, pp. 115–123 (2012)