A Machine Learning Based Prediction System of Medical Laws Judgment Using Statute-Classified Decision Tree with Text Similarity

Yuh-Shyan Chen[1], Hsing-Chen Chi[1] and Meng-Luen Wu[2]
[1]Department of Computer Science and Information Engineering
National Taipei University, Taipei, Taiwan, R. O. C.
[2]Department of Computer Science and Information Engineering
National Taiwan University of Science and Technology, Taipei, Taiwan, R. O. C.
E-mail: {yschen@mail.ntpu.edu.tw}

***Abstract*—Legal affairs assisted by artificial intelligence (AI) has been a new technological trend nowadays. People resort to the law for conflict resolutions, but most of them have no idea about their cases, and do not even know what kind of lawyers they should consult to. Medical disputes occur frequently these days. Doctors are not experts in law, but they care about the outcome of medical lawsuits they encounter. To help people better understand the outcome of a medical lawsuit and prevent complaints abuse, in this paper, we propose a judgment prediction system for medical law cases as a means of preview. We collected 227,494 medical law judgment with their labeled results as our datasets, either for training and evaluation. The medical law related law statue and corpus are the materials for text feature extraction. In our proposed method, there are primarily two parts: the first part is to use information entropy to generate decision trees that predicts the outcome of a judgment; the second part is to use the word2vec method to calculate the similarity distance between a case text and the relative judgments to determine whether a plaintiff is likely to win a case. The experimental result show that the combination of the decision tree and the text similarity achieves 96% accuracy for judgment prediction in medical laws.**

Keywords: Case prediction, machine learning, word2vec, decision tree, text similarity.

I. Introduction

Nowadays, more and more people resort to the law for conflict resolutions, and the number of law cases increases day by day. Therefore, lawyers, judges, and other legal professionals seek for artificial intelligence (AI) solutions to help them solve cases, in order to save time for case analysis and help them make accurate decisions. Currently, there are many existing AI algorithms associated with legal affairs, including natural language processing (NLP) [1]–[9], legal text classification [1]–[4], sentencing prediction [5]–[7], and statute citing recommendation [8], [9].

As medical dispute occurs frequently nowadays, in this paper, we develop a judgment prediction system for medical law cases. The prediction of a judgment is realized by analyzing the relative judgments from the past. Formally, a judgment is comprised of three main parts: the first part is the abstract, which briefly outlines the judge's decision; the second part is facts and reasons, which describes the case in detail; the third part is the law statute, which lists the regulations referenced in the judgment. We use these judgments as materials for machine learning, and the trained model is able to predict the judgment result in the future.

We use the text similarity measurement as a natural language processing (NLP) method for judgment prediction. The text similarity method [2]–[4], [8], [9] calculates the distance between two text vectors through word embedding. The word embedding combines a set of language model and extracts the features from text. In the word embedding, the vocabularies are mapped to real numbers vectors. The vectors are in a feature space with one word per dimension. To improve the performance, dimension reduction is conducted afterwards.

The main contribution of this paper is summarized as follows.

- We propose a statute-classified tree using the statue combinations in a judgment as features, and 85% of the judgments can be completely predicted by the tree generated by this method.
- Find key law statues that decide whether a plaintiff will win or lose in a judgment.
- The statute-classified decision tree with text similarity scheme is able to predict the judgment result of medical law cases, and the accuracy is around 96.7%.

This paper is organized as follows. Section I is the introduction of medical law case result prediction. Section II describes the related work and motivation. Section III describes the system model, problem formulation and basic idea of our proposed scheme. Section IV describes our proposed scheme of statute-based decision tree with text similarity. Section V provides the experiment result, and the conclusion is given in Section VI.

II. Related Work

In this section, we describe the related work of law case result prediction and the motivation of our work.

A. Artificial Intelligence and Law

There are three types of AI-based legal assistant [1]–[9]. The first type is the text classification using machine learning and deep learning [1]–[4]. The second type is the charge prediction using deep learning [5]–[7]. The third type is statute-citing prediction using text similarity [8], [9].

For the text classification, Lei *et al.* [1] proposed a machine learning approach that automatically classifies Chinese judgment documents. A set of judicial terms are constructed for word segmentation. After word segmentation, a judgment document is converted to a vector space using TF-IDF algorithm. The document in the vector space is used to produce a model for text classification. Wagh *et al.* [2] applies network analysis to compare the cosine similarity and citation based similarity between legal documents. Jiao [3] proposed a tree structure-based method to measure the similarity between text contents. The similarity is calculated by combining the similarities in its respective tree layers. Rajshekhar *et al.* [4] improvements the information retrieval on law judgment by adding the diversity and types of semantic relationships between legal corpus,

For the charge prediction, Xiao *et al* [5] proposed several deep learning text classification baselines for charges prediction and relevant law articles citing prediction. In their method, they used the fact part of the judgment as inputs. In addition, the regular expressions are used to extract the applicable law statues, charges and judicial terms in the text. Luo *et al* [6] proposed an attention-based neural network method. Charges of the revalant cases are extracted. Ye *et al.* [7] proposed a label-conditioned Seq2Seq model, which uses the fact description in a criminal case to decode court views conditioned on encoded charge labels.

For the statute-citing prediction, Wang *et al.* [8] proposed a topic model based approach to measure the text similarity of Chinese judgment document, which is based on Latent Dirichlet Allocation (LDA), labeled Latent Dirichlet Allocation (LLDA), and the text feature from judgment document. Hung *et al.* [9] proposed a topic model based approach to measure the text similarity of Chinese judgment document, which is based on the text feature from judgment document using the three-phase prediction (TPP) algorithm and judgment retrieval for statute-citing prediction.

B. Motivation

In recent years, there are many legal AI-related applications, including the text classification [1]–[4], charges prediction [5]–[7] and the statute citing recommendation [8], [9]. These applications combines the traditional machine learning and deep learning technology in field of law. Currently, there is few research on predicting the judgment result. The judge decides the result of a lawsuit according to the fact, reason, the regulations and statue. Particularly, the statute is the foundation of judgment, and the statue pattern cited in the judgement influences the result. In this paper, our proposed system analyzes the case text and finds the key statute and judicial terms that directly affects the result. If not available, our system helps make decisions through the calculation of distance between the text of a case and its revalant cases.

III. Preliminaries

This section describes the system model, the problem formulation, and basic idea in subsections III. A, III. B and III. C.

A. System Model

The system analyzes judgments the through the law statues and corpus. The system is applied to the prediction of lawsuit results in the field of medical law. Our proposed architecture contains the statute-classification, case-based, supervised learning, unsupervised learning, decision trees and text similarity.

The proposed method finds the key statues that affect the lawsuit results. We construct decision trees based on statue as a statue-classified decision tree. If the statue-classified decision tree is not capable of lawsuit results prediction, calculate the text similarity between the case and revalant judgment. The comparison between lawsuit result prediction using text similarity and statue-classified decision tree is illustrated in Fig. 1.

B. Problem Formulation

This main goal of our proposed method is to increase the accuracy by minimizing Mean Absolute Error (MAE). MAE is a measure of difference between two continuous variables, which represents the difference between the expected result $\hat{r^e}$ and the predicted result $\hat{r^f}$. The optimization function is defined as follows:

$$\begin{aligned} &\text{minimize} \ \left\{ \frac{1}{n} \sum_{i=1}^{n} \left\| \hat{r_i^e} - \hat{r_i^f} \right\| \right\} \\ &\text{subject to} \begin{cases} r_i^e \in J, r_i^f \in j^e \\ \hat{r_i^e} \in \left\{0,1\right\}, \hat{r_i^f} \in \left\{0,1\right\} \\ \hat{r_i^f} = \left\{ r_p^1, r_d^1, r_p^2, r_d^2 \right\} \end{cases} \end{aligned} \tag{1}$$

where all judgments J are divided into training dataset j^t and testing dataset j^e, n is number of j^e, $\hat{r_i^e}$, which is defined as the i-th expected result after prediction, whereas $\hat{r_i^f}$ is defined as the i-th predicted result after

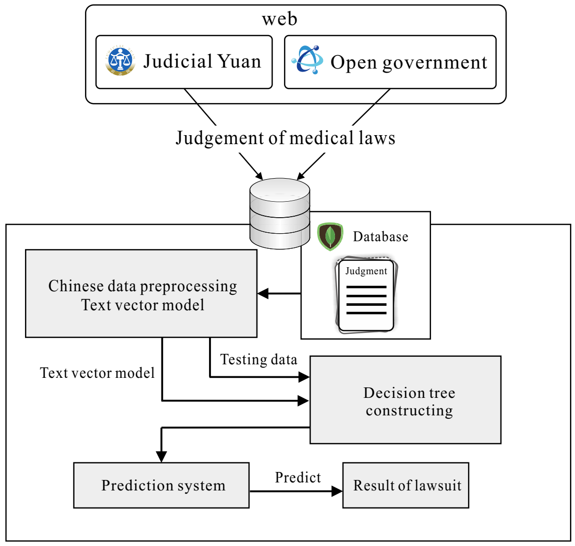


Fig. 1. The System architecture.

prediction, $\hat{r_i^e}$ and $\hat{r_i^f}$ contains two possible judgment results, which are plaintiff wins or loses. If the result after predicted is plaintiff wins, the value of $\hat{r_i^f}$ is set as 1, conversely, $\hat{r_i^f}$ is set to 0. r_i^e is the ground truth. We set r_i^e as j^e for verification. $\hat{r_i^f}$ composed of statute-classified result r^1 and text similarity result r^2 after prediction, the result also contains two possible judgment results, plaintiff wins r_p or loses r_d. The verification of accuracy of the prediction is formulated as follows:

$$\begin{cases} \text{if} & \left\| \hat{r_i^e} - \hat{r_i^f} \right\| = 1, \text{false} \\ \text{else} & \left\| \hat{r_i^e} - \hat{r_i^f} \right\| = 0, \text{true} \end{cases} \tag{2}$$

In the formula, when the difference is 0, the prediction is totally correct; otherwise, the correctness depends on the difference, the less the better.

C. Basic Idea

There are several basic ideas for establishing our proposed medical law case result prediction system. First, extract the features using the statue and law articles in the text. Second, use artificial neural network and deep learning approaches to perform judgment result prediction. In addition, finding the most influential statues is helpful for increasing the accuracy of prediction. Furthermore, classification of cases in advance is necessary for the neural networks to converge. To integrate these idea, we constructed a flowchart illustrated in Fig. 3. First, select law cases of the same type from the dataset. Subsequently, find a series of law statutes as the keys for judgment result classification. The system searches for the keys in a judgment, if not available, use the text similarity analysis through word embedding as a complement.

D. Application scenarios

There are two possible type of users, including professional law workers and non-professionals.

For professional law workers, they can choose some statues and revalant judgments as the initial inputs, and see if it is possible to predict the judgment results from the input materials. There are two phases in the prediction. In the first phase, the system use the input statues and determine whether it is able to use statues alone to predict the judgment result. If not possible, in the second phase, the system use text similarity to compare the input material with the revalant judgments and find the one with the largest similarity and output its result.

For non-professionals, they can take advantage of the proposed method to predict whether they will win the case in the appeal, in other words, they can use their first instance to predict the result of the second instance. There are two phases in this scenario: in the first phase, classify the case type by the statue cited in the first instance; in the second phase, analyze the facts and reasons in the text of the case. Through the two phases, the system predicts the result and gives recommendations to cite specific law statues.

IV. A Statute-Classified Decision Tree with Text Similarity

In this section, an overview of prediction system for medical laws judgment is given. We propose a statute-classified decision tree with text similarity. There are three phases for the proposed method:

1) *Data pre − processing phase*: In this phase, collect the legal judgments from the open data websites. Segment the words and corpus in the judgments, and label the result of the judgments using specific corpus as the clue. Analyze the text using word embedding technique through the segmented text. The word embedding method produce word vectors according to their frequencies.
2) *Decision tree construction phase*: Build a decision tree through the sets of statue groups and text similarity. Find the decisive law statue groups and corpus as the nodes of decision tree. When building the tree, choose the decisive law statue groups as tree nodes first. After all decisive law statues are chosen, use other corpus in the judgment text as the rest of the tree nodes.
3) *Decision tree inference phase*: The last phase is prediction, which also includes two parts. The first part is to use the built tree to predict the result of a judgment. If the case is unpredictable by using the tree, in the second part, use the word embedding method to calculate the text similarity between judgments. The prediction is then obtained from the result of similar judgments.

A. Data pre-processing phase

This phase collects the judgments J of related the medical law judgments from the courts in Taiwan. A

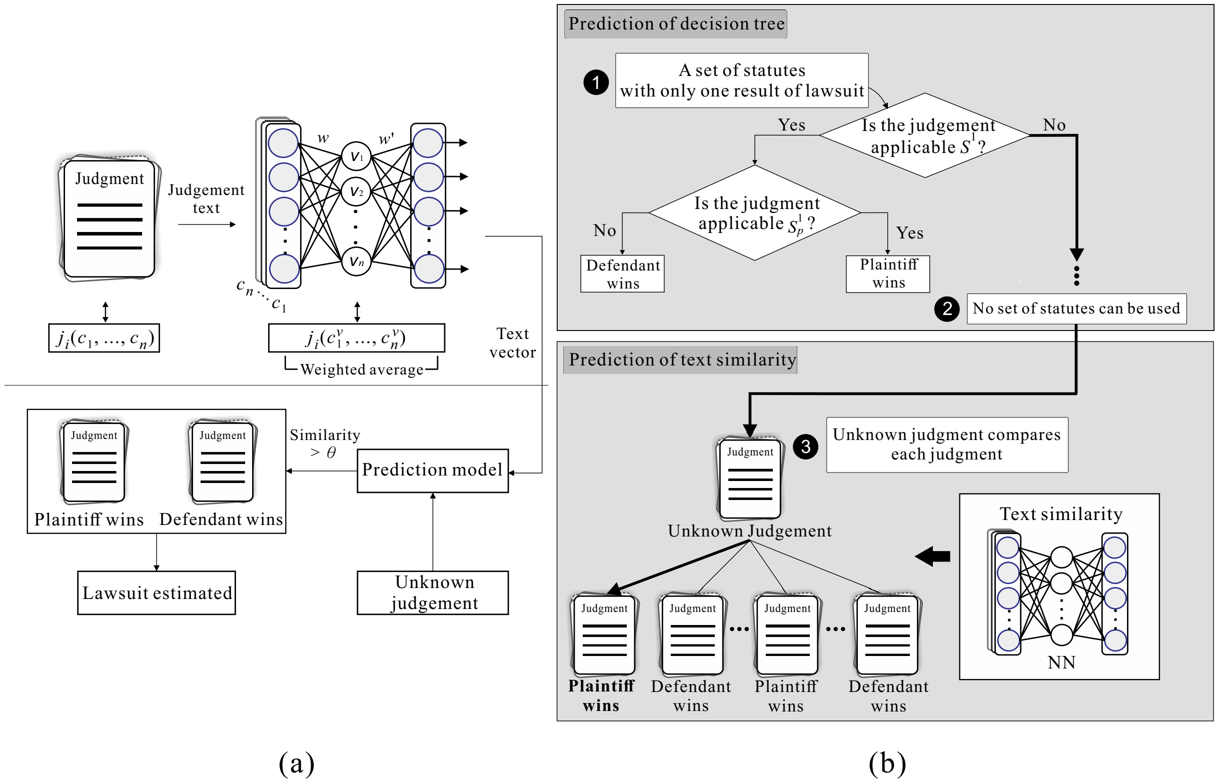


Fig. 2. Comparison of (a) prediction system architecture using text similarity and (b) our proposed scheme using statute-classified decision tree with text similarity.

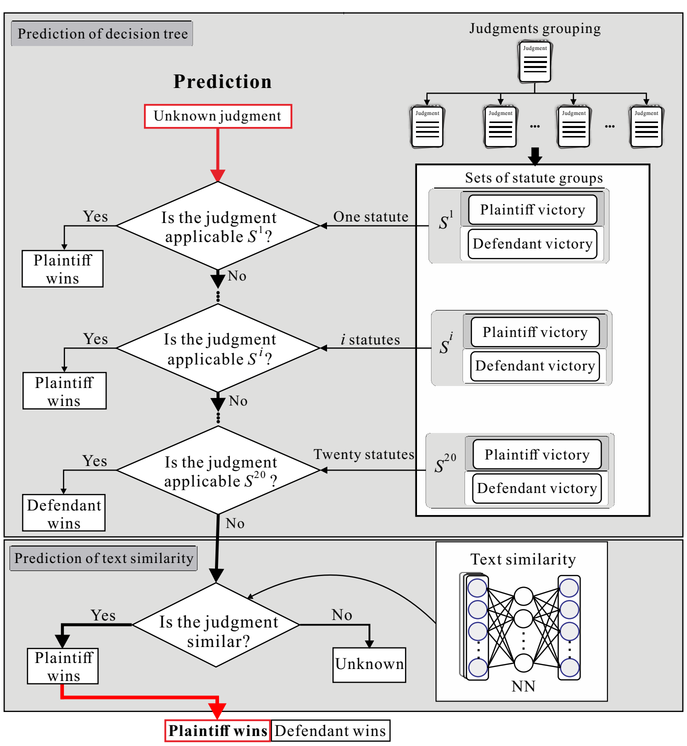


Fig. 3. Basic idea of prediction system

judgment J contains the object, the process, and the statute cited in the law case. The corpus c of a judgment is used to label the result t^o of the judgment. There are three types of results, including the plaintiff wins j^p, the defendant wins (plaintiff loses) j^d, and ambiguous j^u. The text of the judgment x_i is the training dataset $x_i^{'}$ for the word segmentation, the segmented dataset $x_i^{'}$ training text vector model. The detail of the steps in this phase is described as follows:

S1. Collect x_i for medical law cases from Judicial Yuan [10] and Open Government [11] websites, which provide judgment dataset x_i that contains three kinds of information, including main text t^t, the facts and the reasons t^f, and the cited statute t^s. The matrix x_i is expressed as follows:

$$x_i = \begin{bmatrix} j_1 t^t & j_1 t^f & j_1 t^s \\ \vdots & \vdots & \vdots \\ j_i t^t & j_i t^f & j_i t^s \\ \vdots & \vdots & \vdots \\ j_n t^t & j_n t^f & j_n t^s \end{bmatrix} \tag{3}$$

S2. The feature in a judgment x_i is denoted as t^t. The result of a judgment is labeled by analyzing t^t. First, define some key terms in the judgment manually. There are three types of term set for determining the results, which are plaintiff wins, defendant wins (plaintiff loses) and ambiguous. The result of the judgment can be labeled by finding the terms in the text. The judgment result labeling is performed by finding if any types of term set t^t exist in the judgment. If only one type matches, the result is labeled as t^o; otherwise, if t^t contains more than two types of terms, the result is labeled as j^u, which means the result is ambiguous. The judgment with ambiguous result will be discard from the training set.

S3. Segment the words in the judgment use the segementation schemes based on word frequency. The system adds t^s with higher weight to the dictionary. t^f with word segmented is denoted as $t^{c'}$. The function is formulated as follows:

$$j_{(n,w)} = \sum_{i=1}^{m} c_i^n \tag{4}$$

where c_i^n denotes the i-th word of the n-th judgment, w denotes the judgment after the word segmentation. x_i is obtained in the pre-processing step as $x_i^{'}$. $x_i^{'}$ is a matrix expressed as follows:

$$x_i^{'} = \begin{bmatrix} j_1 t^o & j_1 t^{c'} & j_1 t^s \\ \vdots & \vdots & \vdots \\ j_i t^o & j_i t^{c'} & j_i t^s \\ \vdots & \vdots & \vdots \\ j_n t^o & j_n t^{c'} & j_n t^s \end{bmatrix} \tag{5}$$

S4. This steps focus on generating a word vector model through multilayer perceptron, which maps important words in the judgment to a vector. The vector is used to calculate the similarity between words. $j_{(n,w)}$ is obtained from the training result from multilayer perceptron. The word vector c^v is obtained by the pattern of the word embedding. c^v is converted into a text vector T for prediction. Each word vector of $j_{(n,w)}$ is used as the training set of the multilayer perceptron. Each word c uses one hot encoding as the training word sample $c_1^e, c_2^e, ..., c_n^e$. c_i^e is represented as $c_i^e = (0, ..., 1, ..., 0)$. The vector of the objective word c_t^v is predicted through the words of the context c_{t+1}^v, c_{t-1}^v. The context c_{t+1}^e, c_{t-1}^e of the input layer multiplies the initial weight matrix W as c_{t+1}^v, c_{t-1}^v, c_{t+1}^v, c_{t-1}^v, which are the weighted average to get the target word vector c_t^v. The weight W is updated constantly as $W^{'}$ during the training phase. The initial weight matrix function is formulated as follows:

$$W \in \mathbb{R}^{VN} \tag{6}$$

where V denotes the dimension of the word vector space, N denotes the size of the word. Calculate the weighted average of the text vector T, and each word vector c_i^v in the J, which the process is formulated as follows:

$$T = \frac{\sum_{i=1}^{n} c_i^v c_i}{\sum_{i=1}^{n} c_i^v} \tag{7}$$

where T denotes the text vector, the c_i^v denotes the vector for word prediction, the c_i denotes the number of the word.

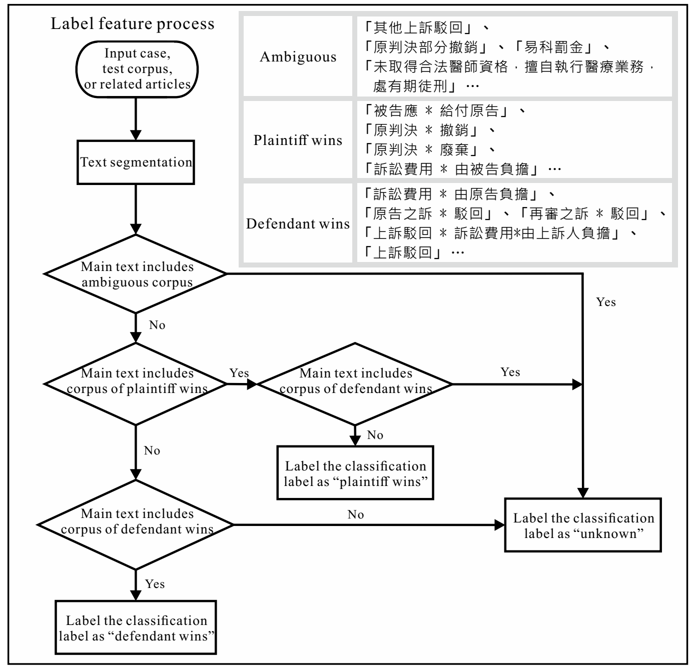


Fig. 4. The data pre-processing phase: feature labelling procedure.

In the dataset, there are 241,905 judgments collected from year 1996 to 2017. As shown in Fig. 4, we analyze the terms c in the main text t^t. There are three types of results in J, including plaintiff wins j^p, the defendant wins j^d, and ambiguous j^u. t^o is obtained by analyzing the result based on the t^t. Assuming that the result is plaintiff wins, t^o is labeled as j^p. When the result is defendant wins, t^o is labeled as j^d. If t^t has the same result of j^p and j^d or meets the ambiguous condition, t^o is labeled as j^u. The number of ambiguous result judgment j^u is 14,411, and is discarded from dataset. t^o within J is denoted as r^e, and r^e is denoted as the number of correctly predicted judgment. The judgment is divided into two parts, and the ratio is 70%:30%. 70% of the judgments are chosen as the training dataset j^r, and the 30% of the judgments are chosen as the testing dataset j^e. The word segmentation is processed by Jieba [12] word segmentation tool, and the segmented result is denoted as $t^{c'}$. As shown in Fig. 5, j^r is obtained from the training process using Continuous Bag-of Words (CBOW) model. c_t analyzed from context is denoted as $c_t = \left\{c_{t-1}^v + c_{t+1}^v\right\}$. c_t^v is weighted averaged denoted as T.

B. Decision tree construction phase

At this stage, the set of the law statues used in the judgment is obtained through feature selection, and the set S and the trained word vector model m^v will be chosen as the decision node of the decision tree. First, the judgments are grouped into G. The law statues are used to classify into multiple sets S in each group of judgments based on

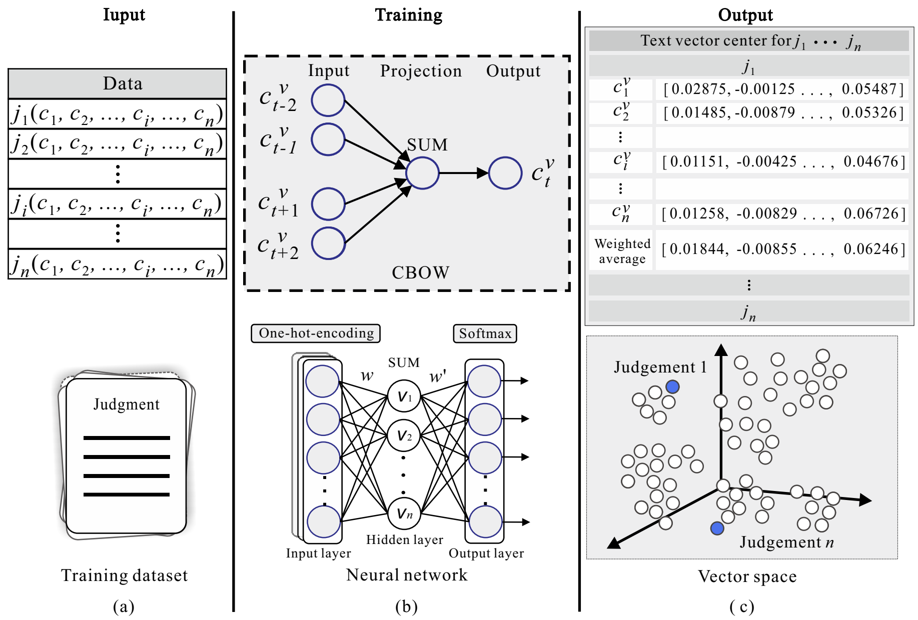


Fig. 5. The data pre-processing phase: training model procedure.

the combination of the law statues and the result of the judgment t^o. Each set is treated as a decision node. When the set S is used up and exhausted, the trained text vector model m^v is used as the final decision node.

Lemma 1: The judgment j of training dataset j^t are grouped G according to the number of statutes in the judgment, the statutes of each judgment is called the law group g. The π-th group G of judgment j is defined as G_π. Total judgments j under the group G_π are classified as sets S^π. The S^π contains three subsets, which are ambiguous set S_u^π, plaintiff wins set S_p^π, defendant wins set S_d^π, respectively. All the law statue groups g under the set S^π are classified as subsets based on the result of the judgment. If the result of a single law case judgment was adopted, the law group g of plaintiff wins was classified as S_p^π, the law statue group of defendant wins was classified as S_d^π, and not a single result was classified as S_u^π. The method will consider the law statue group g, whether all the permutations and combinations in the law statue group g need to be considered together. The all combination of S^π is defined as $\left\{S_1^\pi, ..., S_i^\pi, ..., S_\pi^\pi\right\}$, where S_i^π is denoted as all combinations of the number of statute i. The law statue group sets of plaintiff wins and defendant wins are denoted as S_p^π and S_d^π,

$$\begin{cases} j^t = \left\{G_1, ..., G_\pi, ..., G_{20}\right\}, S^\pi \in G_\pi, \\ S^\pi = \left\{S_u^\pi, S_p^\pi, S_d^\pi\right\}, \\ S^\pi = \left\{S_1^\pi, ..., S_i^\pi, ..., S_\pi^\pi\right\}, \\ \left\{S_1^\pi, ..., S_{\pi-1}^\pi\right\} \subseteq S_u, \\ S_p^\pi \in S_\pi^\pi, S_d^\pi \in S_\pi^\pi, S_p^\pi \cap S_p^\pi = \varnothing \end{cases}$$

Proof: This is an example of proof by contradiction. In our analysis, the $S_1^\pi = \left\{s_1^\pi, ..., s_i^\pi, ..., s_n^\pi\right\}$, s_i^π is the π-th law group under S_i^π. The s_i^π assumes that it belongs to plaintiff wins set, a law statue group of new judgment is denoted as g^n. $g^n = \left\{s_i^\pi, s_1^n, ..., s_i^n, ..., s_n^n\right\}$, where s_i^n is the i-th statute of g^n, with only s_i^π exists in the dataset, and others are new unseen statutes. The g^n predicts the result through the only equal statute s_i^π, and the result is successfully predicted. If s_i^π is not related to medical law cases, which indicates that it is not a statute that affects the result of the judgment. Next, g^n is for defendant wins, and the prediction results will be incorrect according to the previous prediction method. Therefore, the laws statues below π are not used, the set $\left\{S_1^\pi, ..., S_{\pi-1}^\pi\right\} \subseteq S_u$. The S_π^π is the combination used by the proposed method, S_π^π is classified into S_p^π and S_d^π, S_p^π and S_d^π are completely independent. Therefore, $S_p^\pi \in S_\pi^\pi, S_d^\pi \in S_\pi^\pi, S_p^\pi \cap S_p^\pi = \varnothing$. ■

S1. The G are grouped according to the number of t^s in the J. The function is formulated as follows:

$$G = \sum_{i=1}^{20} G_i \tag{8}$$

where G denotes set of judgment grouped according to t^s.

S2. G includes the law group set S, each S has two subsets, which are the plaintiff wins S^p and defendant wins S^d. Because the law statute groups g are the results of single judgment in the S, g will not be repeated in S^p subset and S^d subset. The function is formulated as follows:

$$S = \bigcup_{1 \le s \le 20, 1 \le i \le 2} Ssi \tag{9}$$

where s denotes judgment after grouping, i denotes two subsets.

S3. The decision node n of the decision tree is constructed according to the S. Next, the trained text similarity model m^v is used as the final decision node n.

As shown in Fig. 6, the number of judgment in the dataset is 227,494. The maximum number of statutes s_n is 20, which is denoted as s_{20}. The judgment J are grouped according to the number of statute s_i as $G_i = \{(j_1^i, j_2^i, ..., j_n^i)\}$. Find the g in group G_i as S_i for single lawsuit result. The are two phases in the decision tree generation. The first phase, set the maximum depth of the decision tree to 20, which each decision node n consists of S. Each S has two subsets; one is S^p that combines with s that the plaintiff wins, and another one is the S^d that combines with s that the defendant wins. At the final phase, the m^v as the n_{20}.

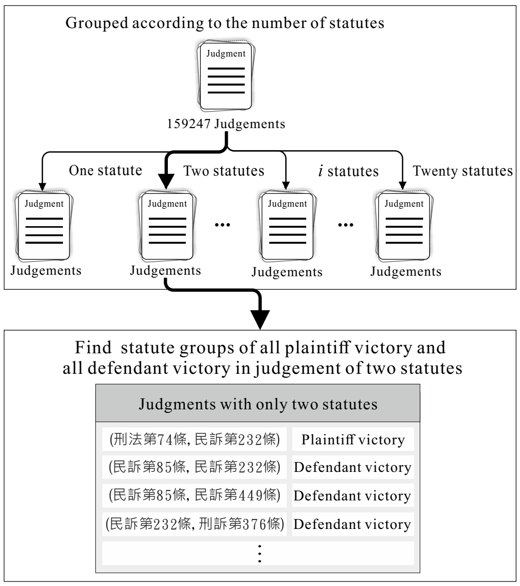


Fig. 6. The decision tree construction phase: law group classifying procedure.

C. Decision tree inference phase

In this part, we use statute and the terms as features for the decision tree generation. There are two steps for generating the decision tree. The first step is to predict by the set S using the law statue group, and the second step is to use the term c in the judgment to calculate the distance between the judgments when there is no law statue group g available for the judge result prediction.

S1. For evaluation, we send unseen judgment j^e as the input of the decision tree. The j^e selects a decision node n of decision tree according to the S. j^e is predicted through S. When using j^e to apply any of the law statue group in the S, the next step is to determine whether j^e is applicable to any sub-node of S^p or S^d. If S^p is applicable, it will be classified as r_p^1, and other applicable S^d will be classified as r_d^1. The function is formulated as follows:

$$r_p^1 = \forall j^e \subset \bigcup_{s=1}^{20} Ss1 \tag{10}$$

$$r_d^1 = \forall j^e \subset \bigcup_{s=1}^{20} Ss2 \tag{11}$$

where r_p^1 denotes the decision tree that predicts the set as plaintiff wins $Ss1$, r_d^1 denotes the decision tree that predicts the set as defendant wins $Ss2$, and j^e is the dataset for evaluation.

S2. When the dataset for evaluation j^e doesn't match any law statue group in S, the second stage of the prediction is carried out, S is used for analysis of text similarity. The function is formulated as follows:

$$r_s^k = \sum_{j \in j^r} log \sum_{k \in j^e} T_{jk} > \theta, r_s^k = r_p^k + r_d^k \tag{12}$$

where r_s^k denotes result of comparison between texts, and $j \in j^r$ denotes the j-th text in the text of the training dataset, $k \in j^e$ denotes the k-th text in the text of the testing dataset. θ indicates the threshold of the text similarity, r_p^k denotes the result of the case for the plaintiff wins in the similar text, r_d^k denotes the result of the case for the defendant wins in the similar text.

The prediction is performed by the following equations:

$$r_p^2 = \frac{r_p^k}{r_s^k} > 0.5, r_d^2 = \frac{r_p^k}{r_s^k} \leq 0.5 \tag{13}$$

where r_p^2 denotes plaintiff wins, and r_d^2 denotes the defendant wins.

The computation of the text similarity function is formulated as follows:

$$T_{jk} = \frac{T_j \cdot T_k}{\parallel T_j \parallel \parallel T_k \parallel} \tag{14}$$

where T_j denotes the text vector of the training data, the T_k denotes the text vector of the data for evaluation.

j^e queries the T of the training model. T_k queries similar judgments T_j is based on the threshold θ. j^e that predicts results based on r_s^k.

As shown in Fig. 7, j^e predicts the outcome of the judgment through S. As long as it meets any decision node of the decision tree, it can directly predict the result of the judgment from tree traversal, and further judge whether the j^e is the r_p^1 or the r_d^1. As shown in Fig. 8, when the decision tree is unpredictable, alternatively predict the result by similarity of the text vectors T. The range of text similarity is the 0-1. Emperically set θ to 0.8, and only consider j of 0.8 or more. Thus, the decision tree predict the result according to the majority of the revalant judgment results.

V. Experimental Results

In this section, the accuracy and effectiveness of the proposed methods are verified. In our dataset, we collected 227,494 medical law judgments with their labeled judgment result. The proposed method is verified by whether it is able to predict the outcome of unseen cases correctly.

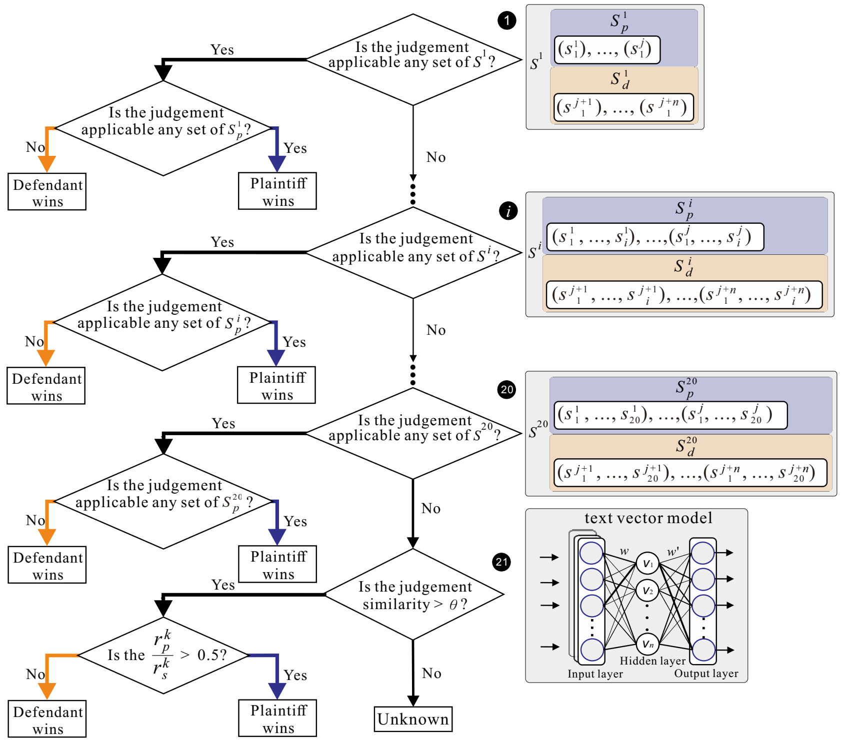


Fig. 7. The decision tree inference phase.

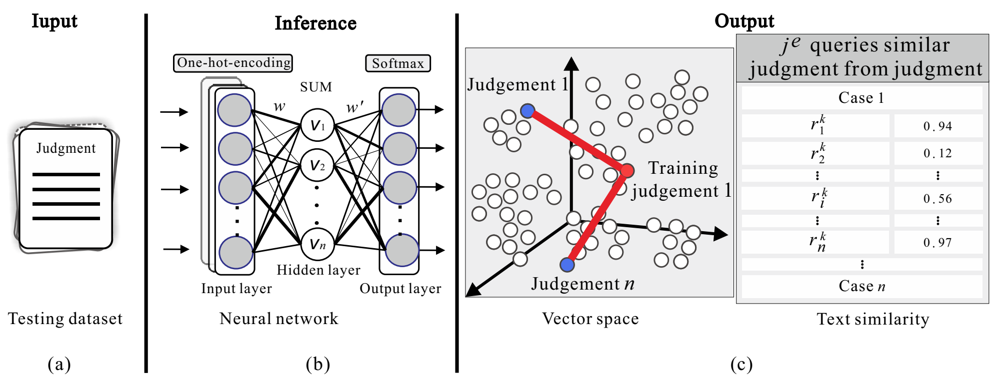


Fig. 8. The text similarity inference.

A. Experimental Setup

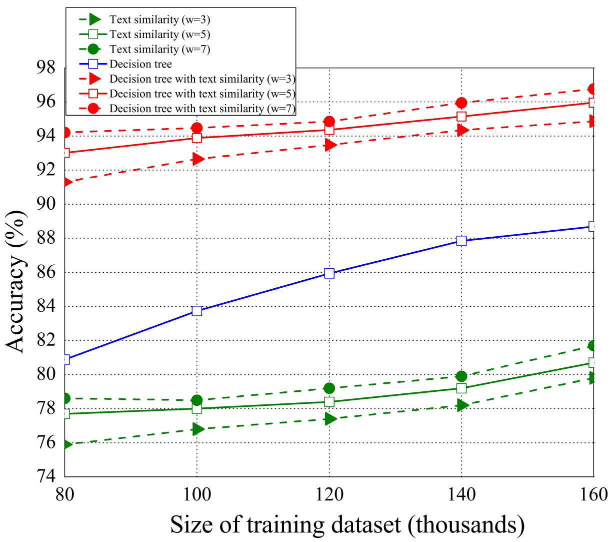
The judgments are partitioned into two sets, which are the training dataset and the testing dataset (dataset for evalution). We use 70% judgments as the training data and 30% judgments as the testing data. The test data are used to evaluate the model built by the training data. In the experiment, there are three algorithms, including decision tree (DT), text-similarity (TS), and decision tree with text similarity (DT-TS), which are drawn in blue, green, and red lines respectively in the following figures. The parameters for the experiment is explained as follows: the parameter of text vector model is *windows* w, w is the number of the context for the text training. TD is the size of the judgment for training. I is training iteration. The accuracy (A) is the percent of correctly classified judgments. The simulation parameter is listed as follows:

TABLE 1
SIMLATION PARAMETERS

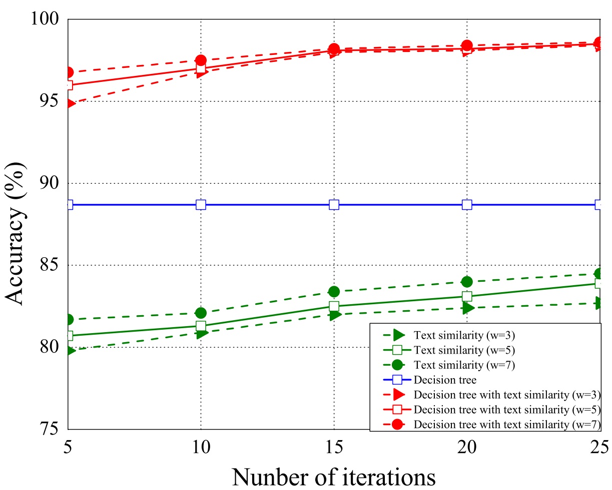
Parameter	**Value**
Simulation tools	Gensim Word2Vec [13]
sg	0
size	250
window	5
mincount	1
iter	5

B. Simulation Results

The simulation results of A under various TDs and Is are shown in Figs. 9(a)-(b), where I is initially set to 5. Fig. 9(a) shows the performances of the TS, DT, DT-TS model in terms of A, under TDs of 80, 100, 120, 140, and 160 thousands of size. For each case, the curve of the A of the DT-TS was higher than that of the DT, and the curve of the A of the DT was higher than that of the TS. In addition, the higher the TD is, the higher the A will be. Fig. 9(b) shows the performances of the DT, TS, and DT-TS model in terms of A, under Is of 5, 10, 15, 20 and 25. For each case, the curve of A of the DT-TS was higher than that of DT, and the curve of A of DT is higher than that of TS. In addition, the higher the TD is, the higher the A will be. Overall, the performance of DT-TS is the best among the three models, which is able to reach 96% of accuracy. DT is second to DT-TS with around 88% of accuracy, and TS model alone reaches around 80% of accuracy.



(a)



(b)

Fig. 9. (a) Accuracy vs. size of training dataset. (b) Accuracy vs. number of iterations.

VI. Conclusion

In this paper, we proposed a judgment prediction system for medical law cases. There are three parts in our method: in the first part, a statue-classified decision tree (DT) is applied; in the second part, the text similarity (TS) between judgment texts is measured by a deep learning scheme based word2vec as a method when DT method is not applicable; in the third part, the combination of the two methods, namely DT-TS method is proposed, which provides the best performance. The simulation result shows that the DT-TS method achieves 96% of accuracy for predicting the result of medical law cases in a judgment. The proposed method is able to give predictions and advices for a medical law related case that helps medical practitioners better understand about their potential problems in law cases, and assist law professionals make better decisions.

VII. Acknowledgments

This work was supported by the National Science Council of Taiwan under Grant MOST 107-2622-E-305-002-CC3, 106-2221-E-305-007-MY3.

References

[1] M. Lei, J. Ge, Z. Li, C. Li, Y. Zhou, X. Zhou, and B. Luo, "Automatically Classify Chinese Judgment Documents Utilizing Machine Learning Algorithms," in *Proceedings of International Conference on Database Systems for Advanced Applications* (DASFAA 2017), pp. 3–17, Suzhou, China, Mar. 2017.

[2] R. Wagh and D. Anand, "Application of Citation Network Analysis for Improved Similarity Index Estimation of Legal Case Documents : A Study," in *Proceedings of 2017 IEEE International Conference on Current Trends in Advanced Computing* (ICCTAC 2018), Bangalore, India, Jan. 2018.

[3] Y. Jiao, "A Method of Calculating Comment Text Similarity Based on Tree Structure," in *Proceedings of 2015 7th International Conference on Intelligent Human-Machine Systems and Cybernetics*, Hangzhou, China, Nov. 2015.

[4] K. Rajshekhar, W. Zadrozny, and S. S. Garapati, "Analytics of Patent Case Rulings: Empirical Evaluation of Models for Legal Relevance," in *Proceedings of the 16th International Conference on Artificial Intelligence and Law* (ICAIL, 2017), London, UK, Jun. 2017.

[5] C. Xiao, H. Zhong, Z. Guo, C. Tu, Z. Liu, M. Sun, Y. Feng, X. Han, Z. Hu, H. Wang, and J. Xu, "CAIL2018: A Large-Scale Legal Dataset for Judgment Prediction," in *arXiv: 1807.02478v1*, Jul. 2018.

[6] B. Luo, Y. Feng, J. Xu, X. Zhang, and D. Zhao, "Learning to Predict Charges for Criminal Cases with Legal Basis," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, Sep. 2017.

[7] H. Ye, X. Jiang, Z. Luo, and W. Chao, "Interpretable Charge Predictions for Criminal Cases: Learning to Generate Court Views from Fact Descriptions," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, New Orleans, Louisiana, Jun. 2018.

[8] Y. Wang, J. Ge, Y. Zhou, Y. Feng, C. Li, Z. Li, X. Zhou, and B. Luo, "Topic Model Based Text Similarity Measure for Chinese Judgment Document," in *Proceedings of International Conference of Pioneering Computer Scientists, Engineers and Educators* (ICPCSEE 2017), pp. 42–54, Quebec City, Canada, Sep. 2017.

[9] L. Yi-Hung, "Judgment Retrieval and Statute Prediction for Legal Problems," *, http://ir.lib.ncu.edu.tw/handle/987654321/66887*, 2014.

[10] JudicialYuan *http://jirs.judicial.gov.tw/FJUD/.*

[11] OpenGovernment *https://data.gov.tw/.*

[12] jieba *https://github.com/fxsjy/jieba.*

[13] gensim word2vec *https://radimrehurek.com/gensim/models/word2vec.html.*