International Journal of Computer Science and Engineering Communications,
Volume.5, Issue.3 (2017): Page.1581-1586
www.ijcsec.com

Legal Text Classification in to Practice Areas Using Machine Learning

Dhrisya V
PG Scholar,
Department of Computer Science and Engineering,
Govt. Engineering College, Thrissur. India
dhrisyav94@gmail.com

Abstract: *Machine learning can be applied for various classification purposes in language processing. Application of machine learning to the legal domain remains a relatively new task. With the increasing ubiquity of the internet, individuals are looking more to internet resources to find relevant attorneys and to obtain answers to their legal questions. In this seminar report, a survey on various classifiers that can be applied for this application is conducted. The steps taken to build a machine learning classifier that successfully classifies legal questions or text into the most relevant practice area is described. We have created 6 different general categories that legal questions fall into. Categorizing legal questions into the correct practice area has many useful applications such as facilitating improved realtime feedback, information retrieval, relevant lawyer recommendations, and responses to users asking questions on Q&A websites.*

Keywords: *Multiclass Classification, Machine learning, Legal text.*

I. Introduction

Machine learning is a type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. Machine learning focuses on the development of computer programs that can change when exposed to new data. The process of machine learning is similar to that of data mining. Machine learning can be applied for various classification purposes in language processing. There are many classifiers that work well for textual data. Application of machine learning to the legal domain remains a relatively new task. With the increasing ubiquity of the internet, individuals are looking more to internet resources to find relevant attorneys and to obtain answers to their legal questions. Legal Q& A websites exist where users ask a question and are given the choice of tagging their question with the relevant topic area. Understanding the different legal practice areas and what they cover can help you locate the right attorney for your needs. Some areas are not of relevance, as they does not deal much with human day-to-day life. To do classification, six popular categories have been chosen, namely Family Law, Personal Injury, Intellectual Property, Criminal Defense, Employment, business Law. Here the output classes are multiple, hence we require multiclass classification. Multiclass classification means a classification task with more than two classes. Mostly used classifiers are binary classifiers, which classify as two only. Therefore, multiclass classification can be implemented in two ways as an OneVsOneClassifier or One-Vs-The-Rest. The meta-estimators offered by sklearn. multiclass permit changing the way they handle more than two classes because this may have an effect on classifier performance. Below is a summary of the classifiers supported by scikit-learn grouped by strategy to have multiclass behaviour.

- Inherently multiclass: Naive Bayes, LDA and QDA, Decision Trees, Random Forests, Nearest Neighbors
- Setting multi class='multinomial' in sklearn.linear model.LogisticRegression.
- Support multilabel: Decision Trees, Random Forests, Nearest Neighbors.
- One-Vs-One: sklearn.svm.SVC.
- One-Vs-All: all linear models except sklearn.svm.SVC.

Multiclass classification conducted using following 5 models are studied here, Logistic regression, Multinomial Naive Bayes(MNB), One layer neural network (1-Layer NN), SVM, A novel semisupervised learning.

The system implementation has many useful applications such as retrieval of more relevant information from the output class, Improved recommendations of relevant lawyers with expertise in the particular topic area, when the question or text is presented to them. Also, Responses to users will be faster, who are asking questions on Q& A websites.

II. Pre-classification Steps[2]

After data for the processing is obtained, pre-processing actions on the data is performed. It make sure that data is fit for efficient classification. Pre-processing steps such as Tokenization, stemming, part-of –speech tagging, parse tree generation will help to yield relevant data alone for further requirements. The feature extraction phase extracts relevant features from the data, so that so that the desired task can be performed by using this reduced representation instead of the complete initial data. The methods used are Word Unigram Features, Word Bi-gram Features, Term Frequency - Inverse Document Frequency (TF-IDF). Another embedding that can give better results is Word2vec. It is a group of related models that are used to produce word embedding's. Word2vec takes as its input a large corpus of text and produces a vector space, typically of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space. Word vectors are positioned in the vector space such that words that share common contexts in the corpus are located in close proximity to one another in the space.

III. Classification

Classification is the task of choosing the correct class label for a given input. In basic classification tasks, each input is considered in isolation from all other inputs, and the set of labels is defined in advance. The basic classification task has a number of interesting variants. For example, in multi-class classification, each instance may be assigned multiple labels; in open-class classification, the set of labels is not defined in advance; and in sequence classification, a list of inputs are jointly classified. A classifier is called supervised if it is built based on training corpora containing the correct label for each input. So, if you are training your machine learning task for every input with corresponding target, it is called supervised learning, which will be able to provide target for any new input after sufficient training. The framework used by supervised classification is shown in Fig. 1.

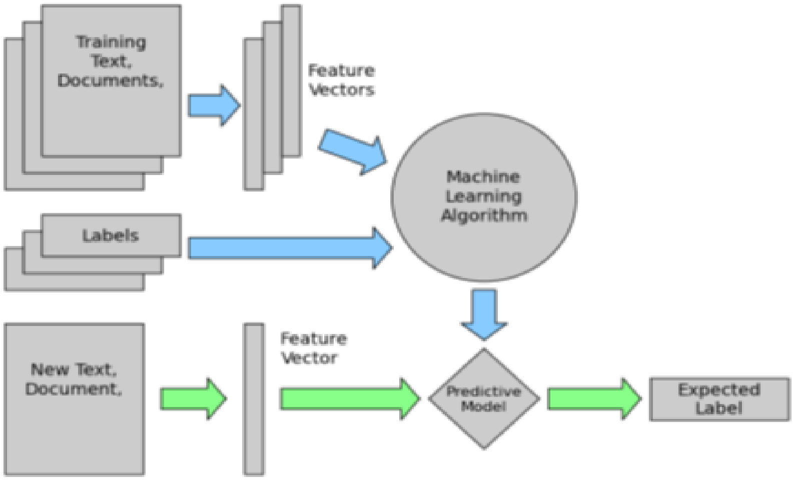

Fig. 1 Supervised Learning Model

III.1 Logistic Regression

Logistic regression, or logit regression, or logit model is a regression model where the dependent variable (DV) is categorical. Cases where the dependent variable has more than two outcome categories may be analyzed in multinomial logistic regression. The logistic function is useful because it can take any real input t; (t € Rt € R), whereas the output always takes values between zero and one and hence is interpretable as a probability. The logistic function σ(t) is defined as follows: LR introduces an extra non-linearity over a linear classifier, f(x) = w^T x + b, by using the logistic (or sigmoid) function, σ().

$$\sigma\left(f(\mathbf{x}_i)\right)\begin{cases}\geq 0.5 & y_i = +1\\ < 0.5 & y_i = -1\end{cases}$$

$$\text{where } \sigma(f(\mathbf{x})) = \frac{1}{1+e^{-f(\mathbf{x})}}$$

III.2 Multinomial Naive Bayes classifier

With a multinomial event model, samples (feature vectors) represent the frequencies with which certain events have been generated by a multinomial (p1,….., pn) where pi is the probability that event i occurs (or K such multinomial in the multiclass case). A feature vector x = (x1,…., xn) x is then a histogram, with xi counting the number of times event i was observed in a particular instance. This is the event model typically used for document classification, with events representing the occurrence of a word in a single document (see bag of words assumption). The likelihood of observing a histogram x is given by

$$p(\mathbf{x} \mid C_k) = \frac{(\sum_i x_i)!}{\prod_i x_i!}\prod_i {p_{ki}}^{x_i}$$

Here MNB model uses Laplace smoothing for maximum likelihood of parameters where we used a Laplace smoothing parameter α = 1.

$$\theta_{yi} = \frac{N_{yi}+\alpha}{N_y+\alpha\times n}$$

In this model, skewed training data will result in a shift of the weights to the biased classes.

III.3 Support Vector Machine[5]

Then, the operation of the SVM algorithm is based on finding the hyperplane that gives the largest minimum distance to the training examples. The optimal hyperplane is computed as follows: The notation used to define formally a hyperplane: $f(x) = \beta_0 + \beta^T x$, Introducing slack variable, $\xi i >= 0$, for $0 < \xi <= 1$ point is between margin and correct side of hyperplane. This is a margin violation and for $\xi > 1$ point is misclassified.

The hyperplane is subject to $y_i(w^T x_i + b)$ _>=1-ξi for all i = 1,…,N; ξ > 0, where xi €R^d is a training example with d dimensions of features, yi € {+1; -1} denotes the label of the feature vector x_i . In this formula, ξ is a positive slack variable, and sum of ξi means the upper bound of training errors. The margin is defined by the distance between two parallel hyperplanes $w^T x + b = 1$ and $w^T x + b = -1$. Therefore, the SVM training process can be defined as an optimize problem as follows:

Minimize $$\left(\frac{1}{2}w^T w + \gamma \sum_i \xi_i\right)$$

where is the regularization parameter.

- _ small value of parameter allows constraints to be easily ignored - large margin
- _ large value of parameter makes constraints hard to ignore - narrow margin
- _ value = 1 enforces all constraints: hard margin

III.4 One layer neural network[4]

A neural network with 35K input units, 100 hidden units, and a softmax function for the output layer is used.

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \text{ for } j = 1,\ldots,K.$$

This model is able to learn complex patterns if a large training data set is available. It is especially useful when we are trying to learn from a sparse feature space. Without current implementation of Neural Networks, the computational time is extremely slow given that we have on the order of 100K input units. One solution to this problem can be to implement and run the neural network on a GPU.

III.5 Semisupervised learning algorithm[1]

An appropriate alternative to the supervised approach is SS learning, which aims to minimize the human-labeling efforts without severely limiting the prediction accuracy. SS learning is a set of statistical machine-learning techniques that exploit a small amount of labeled data and a large amount of unlabeled data for training. The algorithm requires several iterations: At each step, one or multiple labeled sentences are added to the set of positive examples. After that, the learned model is updated. When the most-recent model is applied to the set of unlabeled data, the most-ambiguous unlabeled data points near the decision boundary are identified and presented to a human expert for judgment. The rationale for this decision is that labeling these sentences provides the most amount of information to the weak learner. SS algorithm consists of three steps. The steps are defined as:

1. Initialization: We draw a random sample from the pool of candids that are highly likely to be positive examples according to the established heuristics and ask a human expert to label these examples. We then add a few negative examples as input to the learner. We use the set of positive examples identified by the human plus the set of automatically detected negative examples as the training data to train a first classifier model M.
2. Active learning to find the best candidates for human labeling: At each iteration, train a classifier model M by using the existing training data. Test M on the test data. Select the most-ambiguous candidates predicted to be positive examples, and collect human feedback on them. This is a sequential process, i.e., the hand-labeled data from the current iteration will be used to train M in the next iteration.
3. Automatic label prediction: In each iteration, train a classifier model M with the existing training data, and test it on the unknown dataset, and then select the examples predicted with high confidence into the corresponding training set. This is also a sequential process, i.e., the data predicted and added to the training set in the current iteration will be used to train M in the next iteration. Furthermore, for the first and second steps, we establish the following rule to balance the total amount of positive and negative examples: In each iteration, the same number of sentences that the human labels as positive is also added to the negative set. In order to minimize human-labeling efforts, we need to keep r + l as small as possible. A couple of previous works target to combine active learning and SS learning together, which is equal to combine steps 2 and 3 into one step.

IV. Dataset

The legal based questions and text are available at sites such as[3]:

1. avvo.com
2. ask-a-lawyer.freeadvice.com
3. lawguru.com

Datasets available for download are:

- UCI Machine Learning Repository- Legal Case Reports Data Set
- Australasian Legal Information Institute (AustLII)
- United Nations Convention on the Law of the Sea (UNCLOS)

V. Findings

Neural network implementation tend to be extremely slow on using the 35K input units from the training data word corpus. On testing out alternative input vector forms, filtering was performed and only kept the top K TF-IDF weighted features. A method that incorporates word2vec learn word representations as an alternative method was implemented to reduce the dimensionality of the input feature space while maintaining information about the data. On comparing semi supervised algorithm implemented for automatic definition detection system, the results suggest that as the number of labels provided by human increases, the accuracy of classification rises. On comparing the performance of different supervised learning methods, it gave 95.93% F1 score and 96.72% recall rate for the most-accurately performing algorithm. To minimize human efforts in labeling training data, we proposed and implemented an SS solution that balances prediction accuracy and labeling efficiency. The experimental results show a 90.47% F1 score and 93.44% recall rate, which only costs eight human labels.

VI. Conclusion

In this work a survey on different methods for classifying legal text into practice areas were studied. Many classification algorithms proposed by different researchers are discussed and the issues present in the existing algorithm were identified. Here, the application can be bettered with applying new learning algorithm. When implementing the system, there are chances of errors from various sources. Identifying those and rectifying can improve the accuracy of the system. Also, a comparison of the four supervised method for multiclass classification is compared with the semi supervised technique, which require human effort in labeling data. To minimize human efforts in labeling training data, the SS solution balances prediction accuracy and labeling efficiency.

REFERENCES

[1] Chang, Y., Diesner, J. and Carley, K.M., 2012. Toward Automated Definition Acquisition From Operations Law. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42(2), pp.223-232.

[2] Zahoor, F. and Bajwa, I.S., 2014, August. Automatic Extraction of Catchphrases from Software License Agreement. In Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2014 Sixth International Conference on (Vol. 2, pp. 189-193). IEEE.

[3] Lao, B. and Jagadeesh, K., Classifying Legal Questions into Topic Areas Using Machine Learning.

[4] Goyal, R.D., 2007, November. Knowledge based neural network for text classification. In Granular Computing, 2007. GRC 2007. IEEE International Conference on (pp. 542-542). IEEE.

[5] M. Ikonomakis, S. Kotsiantis, V. Tampakas, Text Classification Using Machine Learning Techniques, Wseas Transactions on Computers, Vol, 4, Iss: 8, pp. 966-974, 2005.