Classification and Verification of Law School Outlines

Arbi Tamrazian
Department of Electrical Engineering, Stanford University
arbit@stanford.edu

Abstract

The ability to automatically verify and classify course related documents is of interest to websites such as CourseHero.com and OutlineDepot.com. In particular, OutlineDepot.com manages more than 55,000 law school outlines that must be manually verified and classified which cause delays and inaccuracies throughout the system. An algorithm that can automatically verify and classify law school outlines will allow for fast information access, increased verification/classification accuracy and reduced costs. Therefore, we have developed a machine learning algorithm that can quickly and accurately verify and classify law school outlines into six major first year law courses. We investigate both supervised (naïve Bayes) and unsupervised (K-means) learning algorithms to validate uploaded law school outlines. To address the classification problem, we have developed and tested three outline classification algorithms that encompass both supervised and unsupervised learning methodologies; multinomial naïve Bayes, Support Vector Machine (SVM), and K-means. We find that our naïve Bayes based algorithms achieve perfect (100%) classification and verification accuracy. However, we find that an unsupervised approach using k-means for both classification and verification provides a dramatic improvement in computational efficiency while only sacrificing less than 1% in classification/verification accuracy.

1 Introduction

The popularity of user contributed content has allowed website like CourseHero.com and OutlineDepot.com to collect, store, and distribute course related documents to students all around the world. These website operate in a "give-and-take" style principle where users upload documents to obtain credits that can be used to download other course related documents. During document uploading, the user supplies supplementary information such as, school name, professor name and course name that the website administrators use to manually verify the document and classify it into its respective category. Only after the document has been successfully verified and classified does the user receive his/her upload credit. Manual verification is a slow process that limits the user's ability to access data in a quick and efficient manner. According to CoureHero.com's FAQ web page, "Our system takes an estimated three days on average for documents to be accepted and credited to your Course Hero account." OutlineDepot.com, which specializes in distributing law school outlines, states that outlines can take as long as one week to be accepted by system administrators. A system that can automatically verify and classify these documents will improve the efficiency of these websites while reducing operational costs.

In this paper, we develop a machine learning algorithm that can automatically verify and classify law school outlines with high accuracy. We have focused on law school outlines in this work because of the availability of training data. Law school outlines consist of 20 or more pages of class notes that are meant to provide a general overview of the important topics of the entire course. Sharing outlines are a popular way for many law students to prepare for final examinations. Rapid verification and classification will allow law students to share law school outlines without delays and will give system administrators an accurate and cost-effective tool to organize course related content. We show that

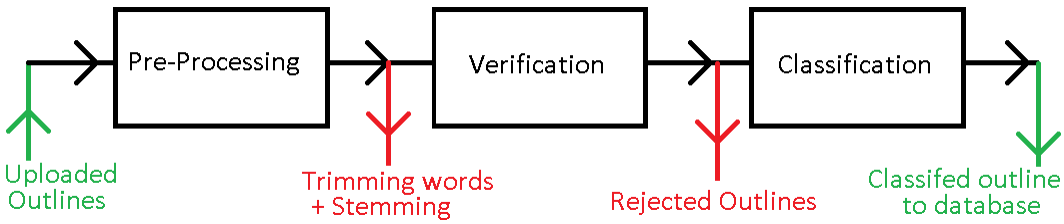

Figure 1: Block Diagram of algorithm: Uploaded documents are first passed to a pre-processing unit which deletes stop words (i.e. with little/no predictive power) and stems all words to their roots. The verification step ensures that the integrity of the website is maintained by rejecting candidate documents that do not fit the pattern of law school outlines. We have developed a verification algorithm that implements a supervised (naïve Bayes) or unsupervised (K-means) approach to validate uploaded documents. Outlines that cannot be verified are removed from the system. Approved outlines are sent to the classification unit where one of three (multinomial naïve Bayes, SVM, K-means) developed algorithms classifies the outlines into one of six core law school courses. These classified outlines are then stored on the database and made available immediately to users for downloading.

a textual classification algorithm using both unsupervised and supervised learning methodologies can achieve fast and accurate outline verification and classification. Although we have focused our attention on law school outlines, the methods presented in this paper can be used automatically verify and/or classify any textual document.

2 Methods and Data

The algorithm is split into three main functional blocks (Figure 1). The first block performs pre-processing on uploaded outlines. The second block determines the validity of uploaded documents by using either a naïve Bayes or K-means algorithm that discriminates law school outlines from other textual documents (not law school outlines). After successfully verification, the classification block can use one of three (multinomial naïve Bayes, SVM, K-means) developed algorithms to classify outlines into six first year core law school courses, which include Civil Procedure, Torts, Constitutional law, Property law, Contracts, and Criminal Law.

We find that with a proper algorithm and feature vector choice we can correctly validate and classify outlines into their respective courses with perfect accuracy (0% test error). All performance measures were conducted using a leave-one-out cross validation methods due to the limited amount of training data.

2.1 Training Data

Training data was donated by law school students and downloaded from various sources on the internet. Due to the scarcity of outlines, we collected 240 law school outlines that encompassed six core first year courses. We also gathered 48 other (non-outline) textual documents to represent a broad range of disciplines (e.g. Science, Art, History, etc). These non-outline training data were used only during training of the verification stage of the algorithm. The training data was manually classified and checked for errors. Table 1 shows analysis on these law school outline training data. Although we achieved high classification and verification accuracy, more training data would be required for refined classification schemes, such as classifying outlines with course name and professor's name.

2.2 Pre-Processing

Extensive pre-processing procedures were developed for simplification and data reduction. Outlines were first converted from Microsoft Word documents to text files using Open Source software[1]. Our algorithm then pre-processed the outlines by: (1) removing all special characters, numbers and punctuation, and (2) removing stop words (i.e. words with little/no classification value, such as

[1] catDOC http://vitus.wagner.pp.ru/software/catdoc/

Course	N =	Five Highest Frequency Root Words
Civil Procedure	36	Court, State, Rule, Claim, Party
Constitutional Law	20	Court, State, Rule, Power, Congress
Criminal Law	22	Crime, Kill, Defense, Intent, Reason
Contracts	56	Party, Perform, Contract, Promise, Breach
Property Law	59	Property, Interest, Title, Easement, Deed
Torts	47	Negligence, Harm, Reason, Risk, Injury
Total	240	

Table 1: High frequency words found in training data.

"is", "and", "but".), and (3) reducing words to their root word using previously developed stemming algorithms[1]. Pre-processing achieved a data reduction of 37% on average.

We computed the frequency of words that appeared in outlines of a specific course (Table 1). This procedure was used to determine the feasibility of the verification/classification problem. The high frequency words from a specific course were well delineated from the high frequency words in other courses, providing evidence that a properly chosen dictionary would allow for accurate classification.

2.3 Verification

To maintain the integrity of course material, we have developed and tested both supervised and unsupervised learning algorithms that can verify uploaded documents by classifying them into one of two categories; "law school outline" or "not law school outline". The verification algorithm attempts to recognize and exploit patterns found in the key words of law school outlines. Users who attempt to submit documents that are not law school outlines (e.g. history essays, art criticism essays, scientific literature, etc) will have their submission rejected since their document will be lacking the necessary pattern of high frequency words found in law school outlines.

One limitation of our current verification strategy is that the algorithm does not check for proper grammar and sentence formulation. This poses a problem for the system since astute users can gain download credits by uploading documents aggregated with a collection properly chosen key words. Such a document would pass the verification step. In future work, we plan to developed specialized algorithms that will assign all outlines a "grammar score" that can be used as a feature variable. This will improve the filtering of illegitimate outlines. The verification step has been separated from the classification algorithm in order to allow for the development of these specialized verification algorithms.

2.3.1 Verification Using Supervised Learning: Naïve Bayes

One of the verification algorithms we have developed and tested is a naïve Bayes (with Laplace smoothing) algorithm that includes the existence of words found in the vocabulary, and word count of the uploaded document as features. The vocabulary, which consisted of 1000 words, was formed by aggregating 500 of the highest frequency words from every positive (law school outlines) training example with 500 of the highest frequency words from every negative (not law school outlines) training example. We find that our naïve Bayes verification algorithm can correct discriminate outlines from non-outlines with a 100% accuracy rate when using this vocabulary.

2.3.2 Verification Using Unsupervised Learning: k-means

We have developed a k-means clustering algorithm that uses the feature vector $x^{(i)} = [f_1,\ f_2, \ldots, f_n]^T$ such that $\|x^{(i)}\|_1 = 1$. Here f_k is the number of times the k^{th} word in the vocabulary appears in the document, $\|\cdot\|_1$ is the L-1 norm, and n is the size of the vocabulary. The L-1 normalization condition is required to account for outlines with different lengths. We find that when using the same vocabulary presented in section 2.3.1, our k-means verification algorithm achieves a 99.5% verification accuracy. This performance metric was measured using a leave-one-out cross validation scheme.

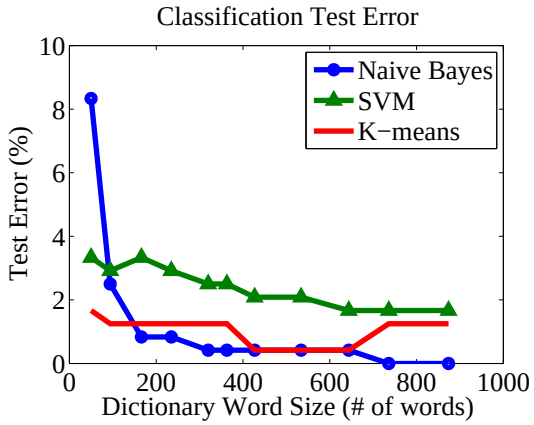

Figure 2: Dictionary Word Size vs Classification Test Error: Here we show how the Dictionary word size effects the classification test error. The test error was computed using leave-one-out cross validation. We find that k-mean dominates the SVM algorithm for all dictionary sizes. However, naïve Bayes is the only algorithm that achieves a 0% classification test error.

2.4 Classification

Verified outlines are passed to the classification block where outlines will be classified into one of 6 first year law courses; Civil Procedure, Torts, Constitutional law, Property law, Contracts, and Criminal Law. Our classification algorithm can utilize both a supervised learning (multinomial naïve Bayes and SVM) algorithms or an unsupervised learning (k-means) algorithm.

2.4.1 Classification Using Supervised Learning: Multinomial Naïve Bayes and SVM

The supervised classification algorithm begins by converting the multicatagory classification problem into a binary category classification problem. For example, when the classification algorithm wants to check if a particular outline belongs to Civil Procedure it will assign Civil Procedure a label of $y = 1$ and all other courses will receive a label of $y = 0$ (or $y = -1$ when using the SVM algorithm). The classification algorithm performs this sub-method on all six course categories until a classification prediction of $y_{predict} = 1$ is found. Although this method is computationally inefficient, we find that it produces a reasonable low test error prediction results.

We used a dictionary that aggregates n of the highest frequency words from all of the outlines in each course category. The resulting dictionary will contain $6n$ words that are not unique[2]. We produce a unique dictionary by removing multiple occurrences of the same word. We find that changing the value of n will change (usually improve) the classification test error for both the multinomial naïve Bayes and SVM algorithm (Fig. 2).

We find that our multinomial naïve Bayes algorithm (with Laplace smoothing) achieves perfect (100%) classification accuracy when using a vocabulary size of 736 words. We show the confusion matrix for the multinomial naïve Bayes using a 94 word size vocabulary in Table 2. We find that 50% of the total error can be associated to confusion between Civil Procedure and Constitutional Law. This was expected since Table 1 shows that there is large overlap between the high frequency words of Civil Procedure and Constitutional Law.

The SVM algorithm achieves a best case test error rate of 1.7% using a vocabulary size of 874 words. We expect the performance of the SVM algorithm to improve as the number of training examples is increased.

[2]the high frequency words in some of the course categories overlap causing a dictionary with words that are not unique. (See Table 1)

		Predicted Class					
		Civil Procedure	Constitution	Criminal Law	Torts	Property	Contracts
Actual Class	Civil Procedure	36	0	0	0	0	0
	Constitution	3	17	0	0	0	0
	Criminal Law	0	0	22	0	0	0
	Torts	0	0	0	47	0	0
	Property	0	0	0	0	59	0
	Contracts	0	0	0	3	0	53

Table 2: Confusion matrix: We have computed the confusion matrix for the multinomial naïve Bayes classification algorithm with a 94 word vocabulary.

2.4.2 Classification Using Unsupervised Learning: k-means

Here we present an unsupervised learning approach using a k-means algorithm which follows the formulation presented by Steinbach et. al.[2]. We follow the same feature vector formulation presented in section 2.3.2. In this implementation, we do not assume six clusters (one for every course category), we treat the number of clusters as a variable in order to correctly model real-life systems. We estimate the number of clusters using the "gap statistic" method presented by Tibshirani et. al. [3]. We find that with a properly chosen vocabulary, the result of the "optimal" number of clusters is always six (as expected).

We find that our k-means algorithm achieves a higher best case classification test error (0.4%) when compared with the multinomial naïve Bayes algorithm, but does better than our SVM algorithm for all vocabulary sizes. We also find that the k-means algorithm becomes unreliable if the dimension of the feature vector is much larger than the number of training data (N=240) (Figure 2).

Although k-means achieves a higher classification test error, it provides many benefits over supervised learning approaches. For example, once clusters are formed, the entire data set can be manually classified for training purposes by manually classifying a single data point in each cluster. Since supervised learning requires every data point to be manually classified for training, this method will provide a dramatic reduction in the cost and time required to manually classify large data sets. Another advantage that k-means provides is computational efficiency. We find that the k-means classification algorithm presented here runs 10x faster than our SVM classification algorithm presented in section 2.4.1.

3 Conclusions

We have developed a machine learning based algorithm to reduce the inefficiencies of law school outline sharing websites. In this paper, we have shown that both supervised and unsupervised learning methods achieve low verification and classification test error (maximum test error of 0.4% and 1.7% respectively). Using a naïve Bayes algorithm for both classification and verification allows for 100% accuracy (in both problems) with a properly chosen vocabulary. However, using an unsupervised approach sacrifices accuracy for a dramatic gain in computational efficiency.

References

[1] M F Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[2] M Steinbach, G Karypis, and V Kumar. A Comparison of Document Clustering Techniques. *KDD workshop on text mining*, 400(X):1–2, 2000.

[3] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society - Series B: Statistical Methodology*, 63(2):411–423, 2001.