Legal relevance classification
a comparison of supervised and unsupervised machine learning methods

Tanja Crijns[a]

Abstract — The field of law accounts for a lot of paperwork, either digital or on paper. Legal professionals show the need to search through their own personal content. A method that can distinguish legal matter from noise is desired. This study compares supervised and unsupervised machine learning methods in legal relevance classification. Three datasets are used; legal documents, newspaper articles and non-official legal texts. The task is to distinguish the actual legal documents from newspaper articles and non-official legal texts. In supervised learning, Naïve Bayes is applied. In unsupervised learning, k-means is applied. Both algorithms achieve an accuracy greater than 0.99. These results are satisfactory. It is however questionable if the results are generalizable due to the suspicion of data leakage.

1. Introduction

1.1 Legal relevance classification

In the Netherlands as well as everywhere else, the field of Law is very important to society. Every day, many juridical cases are dealt with by legal professionals. In the Netherlands in 2015, a total of 1.7 million legal cases were processed by a large amount of judges, chief counselors and lawyers[1]. This large amount of legal cases accounts for a lot of paperwork, either on paper or digital.

Legal Intelligence is a company that has the goal to improve the provision of information for legal professionals. The company has developed a legal search engine that allows these professionals to search a large legal database. This search engine provides professionals with access to legal documents from various public sources. In addition to this, there is an increasing need to search through personal content. Legal professionals often hold their own data repository which contains a combination of legal and non-legal content. Searching through a non-organized repository can take up valuable time. This problem could be diminished by a classifier that can distinguish between relevant legal data and non-relevant, non-legal data.

1.2 Research aim

The goal is to build and evaluate a classifier that separates relevant legal documents from noise. The interest was in which machine learning methods could provide a satisfactory solution to this problem. We were interested to see whether a supervised or an unsupervised method performed best. This interest was precipitated by a thesis on text document categorization(Özgür, 2004). The author compares and evaluates the performance of state of the art supervised and unsupervised techniques for document categorization. Özgür concludes that for unsupervised techniques, k-means and disected k-means perform the best. For supervised, support vector machines achieve the highest performance. For this study, finding a specific method is not the goal, but which field of techniques performs best in general; supervised or unsupervised techniques.

The research question reads:

'How do supervised and unsupervised methods compare on a legal relevance classification task?'

1.3 dataset

Three datasets were provided, a dataset with official legal content, a dataset with Dutch news and a dataset with general legal content. The dataset with official legal content was extracted from the governmental website *www.rechtspraak.nl*[1] and contains 300.000 documents. The datasets with non-legal content are extracted from a Dutch corpus called SoNaR. Among other things, it contains news articles and also legal texts. These datasets contained 700.000 news articles and 7860 legal texts. The SoNaR legal text dataset is used to distinguish general legal text from actual legal documents. The datasets all contains text documents in XML format.

[a]S-number: 4204999. Mail: tanja987.crijns@student.ru.nl
[1]https://www.rechtspraak.nl/

2. Background and related work

This paper reviews supervised and unsupervised machine learning methods for legal relevance classification based on text documents.

Sebastiani reviews all sorts of machine learning methods in text categorisation.(Sebastiani, 2002) He concludes with a ranking of methods, he states that boosting-based classifiers , Support Vector Machines, example-based methods and regression methods perform the best; batch linear and Naïve Bayes classifiers perform the worst.
Yang et al. made another comparison of a smaller set of machine learning techniques such as k-nearest neighbor, Support Vector Machines, Neural Networks, linear least squares fit and Naïve Bayes(Yang, 1999). They conclude that these methods all perform comparably if there are enough documents per category, preferably over 300.

Clustering methods are mentioned but not fully evaluated in the previously mentioned papers. Clustering will be used as an unsupervised method to find the hidden data concept of legal relevance in the data(Berkhin, 2006). Steinbach et al. compare hierarchical clustering with k-means and a bisecting version of k-means(Steinbach, Karypis, Kumar, et al., 2000). They show that k-means outperforms hierarchical clustering in general and that bisecting k-means outperforms the 'regular' k-means.

Machine learning research and specifically text retrieval has been an important addition to the field of law(Moens, 2001). Professionals in this field work with large amounts of text data, machine learning techniques help facilitate and accelerate this process. A lot of research is done on the multilabel text classification of specific law areas.

Mencía and Fürnkranz applied multilabel classification algorithms to a legal database of the EU(Mencía & Fürnkranz, 2010). They evaluated three algorithms in order to tackle the problem of labeling one or more of 4000 labels to a document. They conclude that a pairwise approach has a good predictive performance which could be feasible for large-scale tasks.

Francesconi and Passerini applied Naïve Bayes and Support Vector Machines to classification of provisions in legislative documents. Legislative documents may be seen as a set of provisions. A model of these provisions allows to describe semantics of rules in these documents, which can be used to develop applications and services for legal professional. They achieve satisfactory results. They show that the deviant nature of legislative documents could be used in order to diminish any difficulties for both citizens and legal experts in accessing said documents. (Francesconi & Passerini, 2007)

3. Method

In this section, the tools used to implement the methods will be described. Also, the pipeline will be defined and the implementation of the supervised and unsupervised methods will be discussed.

3.1 Programming language and tools

Most of the applied machine learning methods used implementations of the Python library scikit-learn[2](Pedregosa et al., 2011) with the 2.7 version of Python. A git repository with all code used in this project can be found on Github[3]

3.2 Pipeline

In figure 1 you will find a schematic representation of the pipeline.

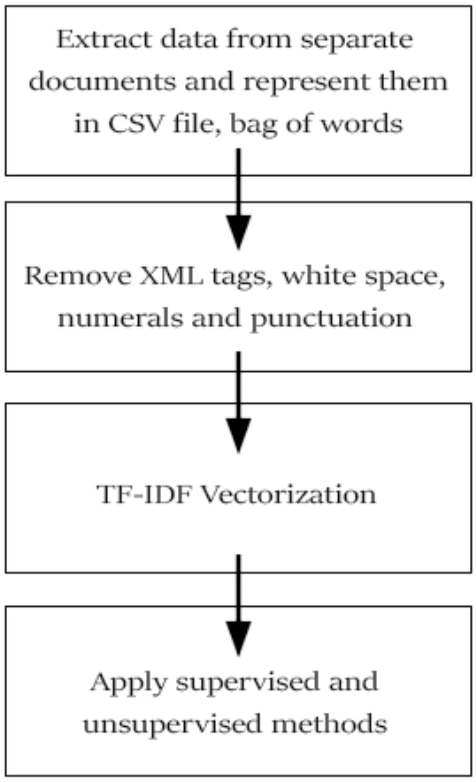

Fig. 1. The classification pipeline

Firstly, the text is extracted from all separate documents in the datasets. Each document was represented as a bag of words.

[2]http://scikit-learn.org/
[3]https://github.com/TanjaCrijns/Text-mining-final-project

As the documents are in XML format, the XML tags and their contents were removed as they have no predictive value over the actual contents of the text. The numerals and punctuation were also removed for the same reason, only the plain words remained.

The word features were transformed with a TF-IDF(Robertson & Jones, 1976) vectorizer in order to get the feature vectors. Lastly, the supervised and unsupervised methods were applied, which will be explained separately. The choice of certain certain methods and how they were implemented will also be discussed.

3.2.1 Supervised learning

Supervised learning makes use of examples of a specific dataset to learn a model of that set. A testset with known labels is used to see how the model performs. As a start, a baseline classifier was used; the Naïve Bayes classifier. This is a binary classifier which is based on Bayes Theorem:

$$P(A \mid B) = \frac{P(B \mid A)\,P(A)}{P(B)}$$

This classifier considers all features to be independent. There was no need for the use of any other classifiers, which will be explained in the results section.

3.2.2 Unsupervised learning

With unsupervised learning, a set of inputs is given without the desired outputs and still a certain structure or relationship between the inputs can be found. An important method here is clustering, where different clusters of input are created and the model will be able to assign new input to a cluster. The k-means algorithm was used as a baseline. This is a centroid based clustering algorithm. A number k is chosen as the desired number of clusters and two centroids are randomly chosen. The algorithm performs two steps; the cluster assignment step and the move centroid step. In the first step, it assigns data points to the closest centroid to form a cluster. In the second step, it moves the centroids to the average of all of the points in the cluster. It repeats these steps until there is no change in the clusters. There was also no need for the use of any other algorithms here, which will be explained in the results section.

3.3 Validation

A k-fold cross validation method with ten folds (Kohavi et al., 1995) was used in order to evaluate the models.

4. Results

4.1 Performance measures

Accuracy is often not the most important performance metric in text classification because of imbalanced datasets. When you have an imbalanced dataset, a high accuracy could be achieved without actually classifying well. For example; if there is a dataset with 99 data points of class a and 1 point of class b. An accuracy of 0.99 can be achieved when only class a is predicted. In this paper however, there is only tested on evenly distributed data. The classifier is also forced to always assign a label, so that it could not be that a high accuracy is achieved without assigning a label to each data point. The accuracy metric will be used for both supervised learning and unsupervised learning. For unsupervised learning, the accuracy was calculated as follows; the number of data points in a cluster that solely belongs to one dataset. Recall, precision and F1-score will not be reported, which will be explained in the remaining part of the results section.

4.2 Supervised learning

At first, 1000 samples of each dataset were used and the intention was to increase the number of samples if the results were not satisfactory. However, at 1000 samples, the following results were achieved:

Data	Accuracy
Legal vs SoNaR news articles	1.0
Legal vs SoNaR legal texts	1.0
Legal vs SoNaR news articles and legal texts	1.0

Recall, precision and F1-score will not be reported as these metrics have no added value when the accuracy is 1.

4.3 Unsupervised learning

Here, also 1000 samples of each dataset were used and the following results were achieved:

Data	Accuracy
Legal vs SoNaR news articles	0.991
Legal vs SoNaR legal texts	0.992
Legal vs SoNaR news articles and legal texts	0.991

5. Discussion

5.1 Data leakage

A performance this high whilst using baseline methods and using only 1000 data points is suspicious. It implies that classification using both supervised and unsupervised methods are nearly flawless. This could imply

that the data differs so greatly that it is easy to classify. However, the suspicion is that it has something to do with data leakage(Kaufman, Rosset, Perlich, & Stitelman, 2012). Data leakage is additional information in the training data which allows the algorithm to produce results that are not realistic. This additional information causes the model to overfit to the dataset, making it not generalizable. Data leakage could present in many different ways, a form of data leakage could be the leakage of test data into the training set or leakage of the correct prediction into the training samples. In the case of this study, there are probably some features that are very predictive for the dataset, but not predictive for the real world problem. The suggestion is that this could have been caused by the XML format that offers more than just plain text. These additional features can be considered metadata, which are not relevant for a text classification problem. These features are often the same or similar for each document in the dataset, but are not necessarily relevant for the actual topic of the document and dataset. Removing some of these recurring features still resulted in a perfect score and it was concluded that it was too hard to manually identify all of the presumably leaked features.

5.2 Future work

The results imply that there is only little left to improve. However, it would be interesting to try this on different datasets. If a similar result is achieved, it is likely that this problem is solvable with baseline supervised as well as unsupervised methods. If dissimilar results are achieved, other pre-processing techniques and algorithms could be tried to achieve a higher performance.

6. Conclusion

In this study, supervised and unsupervised machine learning methods are compared in the problem of legal relevancy classification. Restating the research question of this study:

'How do supervised and unsupervised methods compare on a legal relevance classification task?'

An answer to this is that the baseline supervised Naïve Bayes approach achieves similar performance to the baseline unsupervised k-means approach, both reach an accuracy greater than 0.99. This could imply that the datasets are so dissimilar that this is an easy classification task. The suspicion is however that data leakage could be the reason behind the high performance. Metadata features caused by the XML format could be of such a predictive value that the performance is unrealistic, rendering the results not generalizable. If the results would be generalizable, the performance of both methods would be satisfactory, which would mean that the methods could be used in a real world application of legal relevance classification. As this is uncertain, it would be advisable to try these methods on different datasets to see if the same results can be achieved.

References

Berkhin, P. (2006). A survey of clustering data mining techniques. In *Grouping multidimensional data* (pp. 25–71). Springer.

Francesconi, E., & Passerini, A. (2007). Automatic classification of provisions in legislative texts. *Artificial Intelligence and Law*, *15*(1), 1–17.

Kaufman, S., Rosset, S., Perlich, C., & Stitelman, O. (2012). Leakage in data mining: formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *6*(4), 15.

Kohavi, R., et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai* (Vol. 14, pp. 1137–1145).

Mencía, E. L., & Fürnkranz, J. (2010). Efficient multilabel classification algorithms for large-scale problems in the legal domain. In *Semantic processing of legal texts* (pp. 192–215). Springer.

Moens, M.-F. (2001). Innovative techniques for legal text retrieval. *Artificial Intelligence and Law*, *9*(1), 29–57.

Özgür, A. (2004). *Supervised and unsupervised machine learning techniques for text document categorization* (Unpublished doctoral dissertation). Citeseer.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Robertson, S. E., & Jones, K. S. (1976). Relevance weighting of search terms. *Journal of the American Society for Information science*, *27*(3), 129–146.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, *34*(1), 1–47.

Steinbach, M., Karypis, G., Kumar, V., et al. (2000). A comparison of document clustering techniques. In *Kdd workshop on text mining* (Vol. 400, pp. 525–526).

Yang, Y. (1999). An evaluation of statistical approaches to text categorization. *Information retrieval*, *1*(1-2), 69–90.