# Legal Citations Classification

ITEC873 Project Report - 41895118 - Ka Yu Lau, ka-yu.lau@students.mq.edu.au

## Abstract

In order to use legal citation to summarize legal documents, it is important to classify whether the citation is accepted by the court or not. Previous work are done to solve this problem. In this paper, we expand investigation on more type of conventional machine learning algorithms and techniques as well as newer machine learning Convolutional Neural Network to solve this problem.

Although simple conventional machine learning tends to overfit and failed to classify the minority class, we used voting and achieved similar performance to previous work which has domain expert involved.

Due to the limit of data size, the convolution neural network failed to pick up the minority class. More data and data augmentation is suggested for future study.

## Introduction

Information overload is a problem in the legal field. Legal professional has spent hours in reading documents. Effort has been made to summarise legal documents in the past two decades. However, due to longer average length, the style of writing and the convoluted and philosophical nature of legalese, legal documents were seen to be more difficult to classify compare to scientific article and news (**Hachey and Grover** )

In 2010, UNSW researchers, Galgani and Hoffmann, create a system called 'LEXA'(Legal tEXt Analyzer). The system aimed to reduce legal professionals' information overload by summarizing information in case reports. One of the appoarches LEXA used to acquire knowledge from the documents is through citation classification.

Based on *Stare Decisis* in common law, judgement is based on previous cases and decisions on how the law was interpreted. Past decisions has a binding effect on following decisons. Court decisions or cases can be instructive as they introduce a new principle or rule, modify or interpret an existing principle or rule, or settle a question upon which the law is doubtful. Galgani and Hoffmann suggested: by classiying legal citations in the case document, we extract a important information about a legal document which helps the summarisation of legal document.

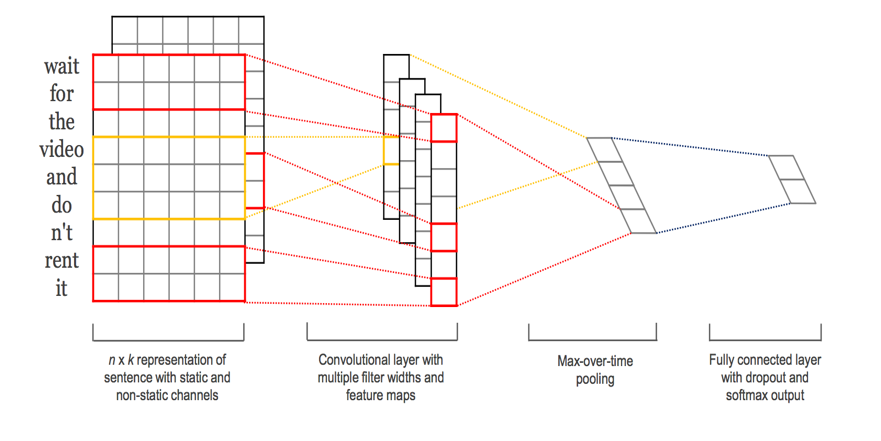
This paper aim to extend the prior work by Galigani and Hoffmann with LEXA, which used Rule-based method, to other machine learning based method and convolutional neural network.

## Related Work

**Legal Citation Classification**

As mentioned in Galgani and Hoffmann's paper, LEXA's classification rules are created under the monitor of a domain expert and based on regular experssion in the text. Expert made corrections to the rule when error occurs in building their knowledge base. This incremental refinements method is called Ripple-Down Rule (RDR) (Compton and Jansen,1990). Although domain expert have not involved in our project, we adopted their approach in measuring the performance of the model such as precision, recall and F1-measure and used their result as a benchmark of our models.

**Multi-Channel Convolutional Neural network**

In Yoon Kim's paper “[Convolutional Neural Networks for Sentence Classification](https://arxiv.org/abs/1408.5882).”(2014), Kim experimented with multi-channel convolutional neural network by processing different n-grams at a time. A text was converted to different n-grams embedding and feed to the model as different channels as similar to image classification problem. The initiative is to prevent overfitting due to small dataset. Although Kim experimented with static and dynamic (updated) embedding layers, we simplified the approach and instead focus only on the use of different kernel sizes.

## Goal of this Project

The goal of this project is classifying whether the legal citation is accepted or rejected by the court. The definition of citation is as followed:

* **Allow** - A principle of law articulated in the primary case is applied to a new set of facts by the court in the subsequent case.
* **Follow** - The annotation is similar to applied but is used in circumstances where the facts in the primary case resemble reasonably closely the facts in the subsequent consideration case
* **Distinguished** - The court in the subsequent case holds that the legal principles articulated by the primary case (usually otherwise persuasive or binding authority) do not apply because of some difference between the two cases in fact or law

To simplify our classification problem and to be compared to Galgaini's previous work, we have combined the groups into "**Follow/Allow**"(FA) or "**Distinguished**"(D).

## Data

**Sample of the dataset**

This dataset contains Australian legal cases from the Federal Court of Australia (FCA). The data included 3890 cases from the year 2006, 2007, 2008 and 2009. Citation classes are indicated in the document, and indicate the type of treatment given to the cases cited by the present case.

**Distribution of the data**

Data is labeled by Galgani's team. We only use the subset of citations summary which labeled with the two interested classed. Our final dataset contains 3956 citations (460 in class D and 3496 in class FA) which is severly imbalanced.

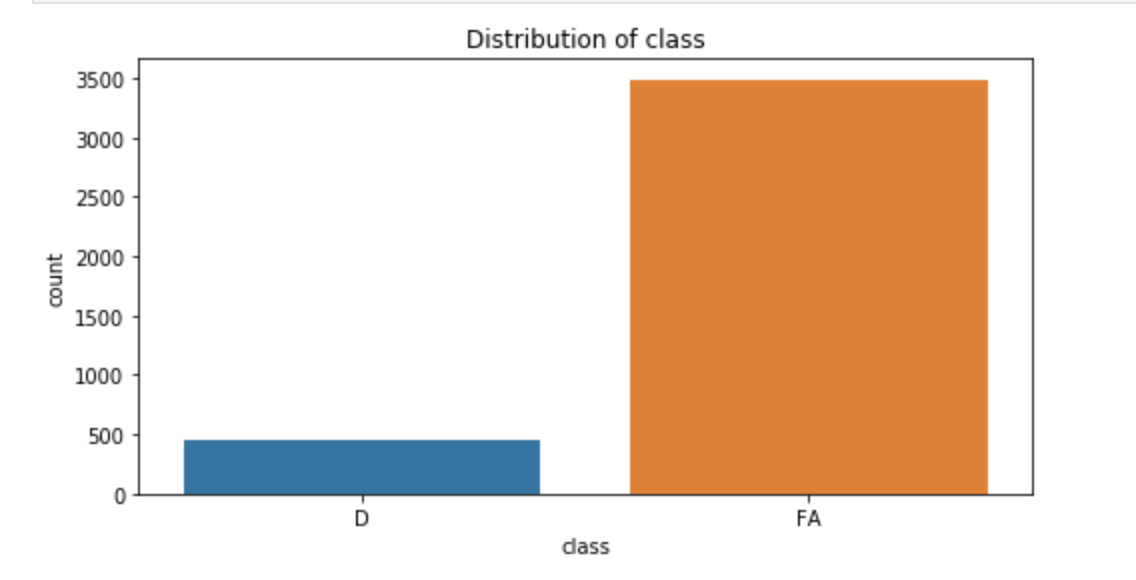


Figure 1 multichannel CNN, Kim(2014)

Figure 2 Distribution of citation class

**Train-test-split**

The 2016 cases (1427 citation summaries) will be used as test set. The remaining (3950 citation summaries) will be used as our training set.

## Metrics

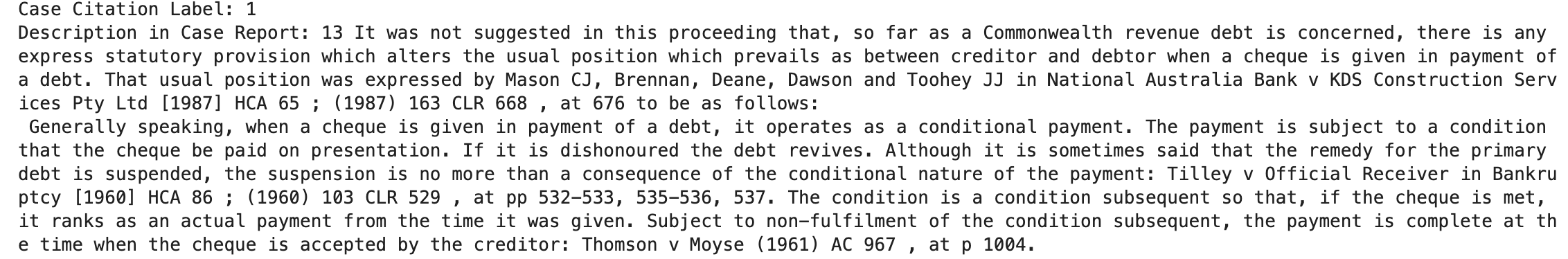
As the data is serverly inbalanced, result can show high accuracy (>80%) by classfying all the observations to the majority class. And this overlook the Type II error. Instead of viewing accuracy only, we will look into precision, recall and F1-measure of each class.

Figure 3 example of citation summary

According to Galgani's paper, it is indeed more important find the distinguished citation in the study, we will evaluate our model based on how well the model classify class D.

## Methodology

### Data Preprocessing

**Upsampling**

As discussed earlier, the data set is severely imbalance. In order to increase the weight of the minority class, we applied naive random-oversample to oversample the minority class. Based on our grid search result, we oversampled the minority 30% of the majority class.

**Feature Exraction**

For feature extraction in conventional machine learning, we used the stop words list from spacy to remove the common words in the text. Then the text will be breakdown into tokens. As we would like to experiment the effect of different features, we first calculate tf-idf, count and occurence for bag of words. Then we expand that to 2-gram, 3-gram and mix of unigram, 2gram and 3 gram.

Due to the length of the text, it is possible to have a word matrix with features more than the observation. To avoid the curse of dimensionality, we limited our word vector to only having 1000 features.

## Model

According to Galgani's paper, LEXA used Ripple Down Rules (RDR) methodology to create a set of conditional rules with the domain expert to classify citations. He also used Naive Bayes Classifier as a benchmark for comparison. In our experiment, we first explore to the effect of other the conventional machine learning technique. Then we will use the convolutional neural network proposed by Kim to compare deep learning with converntional machine learning algorithms. All parameter of these model are chosen based on result from grid search (for detail result please refer to the conventional machine learning notebook)

### Conventional Machine Learning

**Multinomial Naive Bayes (MNB)**

As mentioned, multinomial naive bayes is used by Galgani. Assuming different class citation summary uses different vocabulary, multinomial naive bayes applied bayes theorem to classify the document. It takes the probability of token occured in the document(prior distribution) to estimate the probability of which class the citation belongs to (posterior distribution).The Additive (Laplace/Lidstone) smoothing parameter Alpha is set to 0.7.

**Decision Tree (DTC)**

LEXA used decision rule based on regular expression to spit the data into two classes. Decsion tree used a similar approach by spitting nodes based on the statistic of a feature. As some words can appears a lot while some do not, the frequency of token can be very different. Since decision tree is insensitive to extreme values, it is suitable for this scenario. To avoid overfitting, we have set the maximum tree depth as 12.

**Logistic Regression (LOG)**

Treating each word as a predictor in a logistic regression model, logistic regression estimate how much the occurence of certain word increases the chance of the document being one of the group.

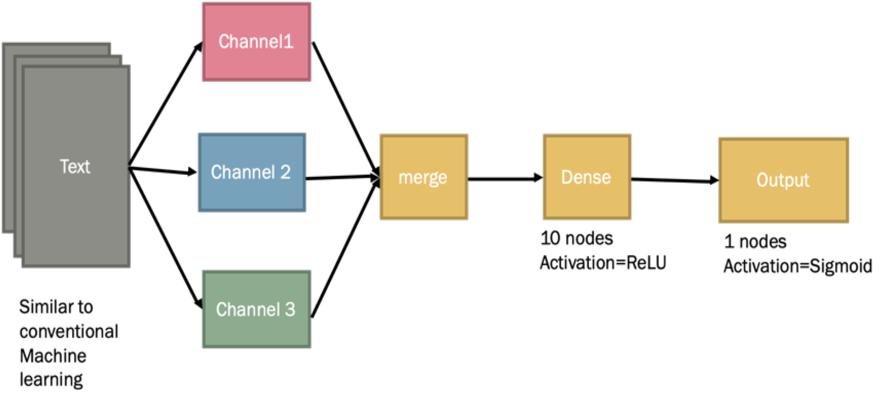
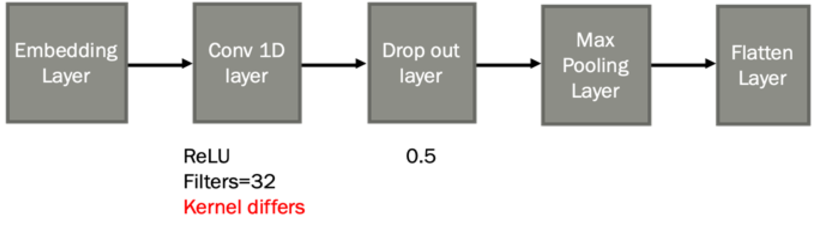


Figure 4 Our multichannel CNN architecture divided text into 3 channels

Figure 5 for each channel we have a one layer CNN architecture

**Support Vector Machine (SVM)**

Consider each of the word as a dimention, support vector machine tries find a multi-dimensional plane that can seperate the two classes. This assumption certain word can identify class is similar to picking up the conditions in the text that LEXA used.

**K-nearest Neighbors (KNN)**

We assumed if two documents used similar words are more likely to belong to the same class. K-nearest Neighbour classified classify based on which class of the closest K observations of the new observaton belongs to. We chose k to be 27 which means the 27 closest obsercation will be considered.

**Ensemble Method: Voting**

We founded the models we create from the basic technique tends to overfit and produce low prediction for the minority. We tried essemble methods that combine the results of weaker models to provide a stronger model. The technique we used is called voting. Each of the model provides a result (vote) which class the citation belongs to. The class get the most vote wins.

### Convolutional Neural Network

For convolutional neural network, we used a multi-channel architecture (figure 4). By considering different number of words is similar to different color in image classification, we passed the text to 3 different channels. Each channel contain an embedding which convert the text to word embedding (figure 5). Then it is followed by a 1-dimentional convolution layer that is different for each channel. A specific kernel size (4, 6 or 8) is assigned to capture different numbers of words. A drop out layer is then added to randomly drop out 0.5 of the data to calculate the weight of each node. The output is then passed to a Max Pooling layer which pick out the maximum value of each pool. Result is then flatten and merge into a single matrix. This matrix will then pass to a simple neural network with sigmoid function to calculate the probability for which class the text belongs to.

## Result

### Conventional Machine Learning

#### Comparing different algorithm

Using occurrence as our features, Table 1 shows the training and testing performance of the models trained by different algorithm compare to LEXA and G\_NB4s(Naïve Bayes Classifier based on sentences by Galgani). All the models showed high accuracy as expected. MNB and KNN’s F1\_measure(D)(F1\_D) is low shows that their performance on classifying the minority class was poor. SVM classified the minority class with a high precision but it didn’t capture only a few of the class D citation (low recall). Decision tree and logistic regression shows a good result in training set but they all overfitted the training data and performance dropped significantly in test data.

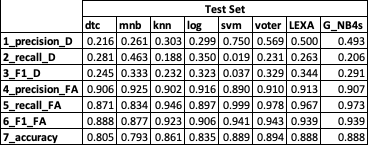
However, by combining all these weaker models with voting, we build a model which performed as well as LEXA. The precision of our ensemble model is better than LEXA but it did not capture as much minority class as LEXA did.

Table 1 Train and test result for different conventional machine learning algorithms

#### Comparing different features

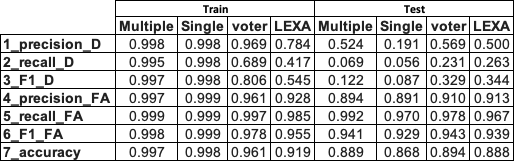
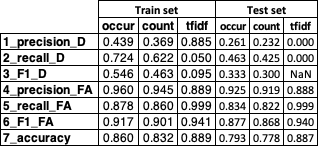
Using MNB as our algorithm, we set up a test on different type of features. Although tfidf performed the best in the training stage, it showed a low recall(D) which means the model cannot capture the minority class despite having a high precision. As result, occurrence of a particular word is a better feature to use. This also shows that the occurrence of a particular word is more important than how many times the word appeared.

Table 2 performance of different features

Table 4 performance of multiple and single channel CNN

#### Comparing different n-grams

For number of grams, unigram worked the best and as number of word per token increased, the model has a poorer performance. However, a potential reason for the result can be result from us removing the stop word in data preprocessing which destructed the meaning of multiple words.



Table 3 performance of different number of words in a token

### Convolutional Neural Network

Both our single channel and multiple channel CNN model performed better than LEXA and our voting model in training but failed to capture the minority class in the dataset in testing.

From the plots, we saw that the validation accuracy did not change much with the number of epochs but the loss of the validation had. For single channel, loss increased linearly after the 3 epoch which indicated the model is overfitting. On the other hand, multiple channel CNN did not showed any trend. This shows that constructing CNN model with multiple channel had increased the generalization of model and solved the overfitting problem.

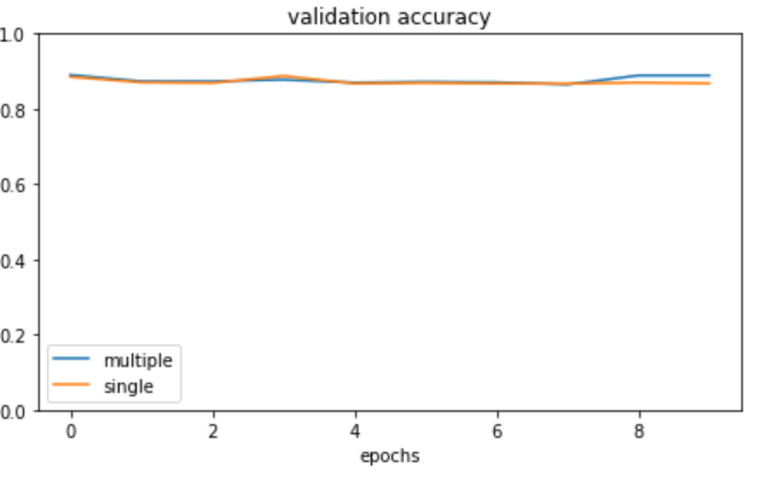
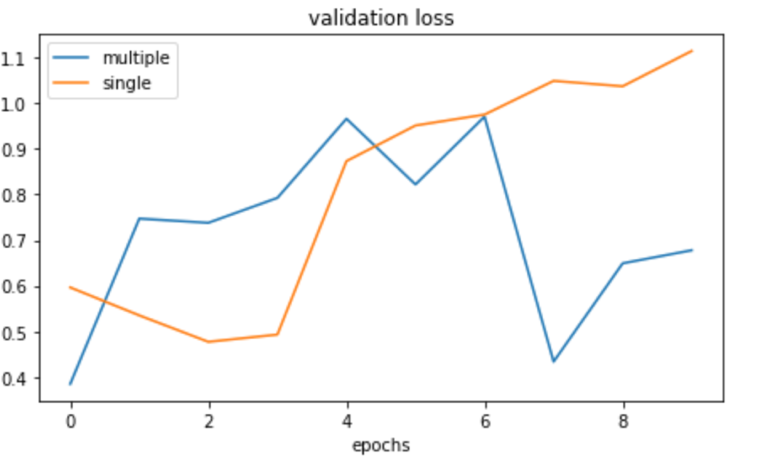


Figure 6 validation loss and accuracy of CNN model

## Further work

To deal with the imbalance of the data set, we adopted naïve random oversampling to solve this issues. However, oversampling did not provide more information about the minority class. This leads to all the models failed to capture class D citations. In future study, we can explore under-sampling technique to combine majority class observations so the features that both class shared can be less overlap.

As applying ensemble technique shows some performance improvement, future study can be focus on adjusting the weight of the weaker models and even exploring other ensemble method such as boosting.

For CNN, as our data set is small, the model did not have a good performance especially in classifying the minority group. Future study should be focus on increasing the sample by data augmentation. As the multiple channel architecture shows a generalized model, we can increase the number of channel and feed other type of features in future study.

Another approach to solve the lack of data issue is to by transfer learning which utilize pretrained model such as doc2vec. Although the document trained doc2vec may be different from legal, retraining some of the layers we may help to overcome the issue.

Other type of text features such as sentence level, character level or tagging are not used in this study.

Future work can be done to exploring the effect of these features to the classification result.

## Conclusion

In this report, we followed Galgani and Hoffmann’s work in 2010 on legal citation classification (LEXA). We explored conventional machine learning technique Galgani did not used such as Decision Tree, Logistic Regression, Support Vector Machine, K-Nearest Neighbor as well as ensemble techniques that combine weaker model in to a stronger one. Although simple conventional did not perform as good as LEXA, the ensemble model achieved similar performance by combining the vote of each models to make a classification.

We also compared a multiple channel CNN with single channel CNN and conventional model. Multiple Channel CNN provide a more generalized model than single channel and avoided overfit. However, due to the lack of data, the CNN models failed to classified the minority class. We suggested future study can be performing data augmentation on the data to create more samples.

## References

Galgani, Hoffmann, 2010, LEXA: Towards Automatic Legal Citation Classification

Galgani, Compton, Hoffmann,2015, LEXA: Building knowledge bases for automatic legal citation classification

Yoon Kim (2014), “[Convolutional Neural Networks for Sentence Classification](https://arxiv.org/abs/1408.5882).”

Paul Thompson, 2001, Automatic Categorization of Case Law

Harry Surden, Machine Learning and Law, 89 Wash. L. Rev. 87 (2014), available at http://scholar.law.colorado.edu/articles/81