Supervised Legal Area Classification of French Judgments

I. Problem Description

This project intends to use deep learning in order to classify French Legal Cases into Private Law Legal Areas.

A. Problem definition

Informal definition

The different legal areas are determined according to the nature of the parties involved. We could classify the private law into four classes, as follows :

- **Civil law** is the general branch of private law which applies to relationships between natural or legal persons.
- **Commercial Law** is the branch of private law which applies to trade relationships.
- **Criminal Law** is the branch of private law which applies to relationships between persons and the society.
- **Labour Law** is the branch of private law which applies to the relationships between an employee and its employer.

Formal definition

We denote the set of legal areas as L, with $|L| = 4$

We can classify a given case c_i into a legal area $l \in L$ if the attributes of the case $\vec{v}c_i$ are implicated by l, where:

- c_i is a given case
- l is a Legal Area (eg : Civil Law)
- $\vec{v}$ is a vector of attributes

Cases may fall into more than one legal area but never none.

Thus, our goal is to learn $f_\cdot : \ c_i \mapsto L_{c_i}$ where . denotes optimality. In order to accomplish this goal, we need to identify the set of attributes of each legal areas $\vec{v}$ and to compute their importance.

B. Proposed solution

It follows from the previous definition that we need to define the data required ($\vec{v}$), the appropriate algorithm (f) and the metric we seek to attain the maximum value of ( ).

Classification

We treat this as a supervised text multi-class classification task. Given a feature vector that describes the text content of a case,we must output a vector of a lenght equal to the number of predicted class which accepts two values 0 and 1, where 1 means that the corresponding label applies to that text and 0 means otherwise.

Data required

A legal analysis of this problem might suggest that we use the facts, parties involved, and procedural trail of the case under examination. However, this type of dataset does not exist

so we will otherwise use the semantics (the vocabulary and the semantic latent space) of the written transcription of the cases, namely, the judgments.

Metrics

In this context, optimality is defined as the trade-off between the number of cases correctly classified and the number of cases correctly retrieved, formalized as

$$precision = \frac{\{number\ of\ correctly\ classified\ cases\} + \{number\ of\ correctly\ retrieved\ cases\}}{\{number\ of\ correctly\ retrieved\ cases\}}$$

$$recall = \frac{\{number\ of\ correctly\ classified\ cases\} + \{number\ of\ correctly\ retrieved\ cases\}}{\{number\ of\ correctly\ classified\ cases\}}$$

Given the precision and recall measures, we need a metric that can measure the aggregation of the previous two. We chose to compute this metric as the (harmonic) mean of the previous two, also known as f-1 score. Since the harmonic mean can not exceed 1, the trade-off is deemed sufficient when we can not find a value closer to 1.

II. Creating the Dataset

The data is provided by data.gouv.fr (found here : https://www.data.gouv.fr/fr/datasets/inca/) and is publicly available. It consists of ≈ 400 000 unpublished judgments (not published in the Bulletin) from the french *Cour de Cassation* since 1989.

A. Collecting the dataset from the datasource

The dataset can be accessed via ftp without authentication, which made it easy to collect data. It is comprised of multiple small cluster of files and a big file (Freemium..) compressed in tar.gz. We scripted the download of individual files with a python script provided by the community and the Freemium file with the wget UNIX command.

The collected data can be found in /data/raw/output.csv

B. Inferring the target labels

However, our dataset does not comprise law areas, so we have to infer them. In order to achieve our goal, we used the material organization of the Cour de Cassation which can be decomposed as multiple chambers which each have its own competencies. Given that each judgment is annotated with the formation which ruled it and that each formation maps to a legal area, we can infer the required labels.

However, the Cour de Cassation has a lot of occasional formation (Commission réparation détention, Chambre mixte, Assemblée plénière) which data is not sufficient enough to have an interest, so we dropped them.

Chambre	Competencies	Corresponding Legal Area
Chambre civile 1	Mariage Law, Inheritance Law, International Private Law, Adoption Law,...	Civil Law
Chambre civile 2	Civil procedure,...	Civil Law
Chambre civile 3	Real estate Law,...	Civil Law
Chambre pénale	Criminal Law, Criminal Procedure Law,...	Criminal Law
Chambre sociale	Collective labour law, Individual labour law,...	Labour Law
Chambre commerciale	Competition Law, Bankruptcy Law, Banking Law, Industrial Property Law,...	Commercial Law

Since we aim to predict general law areas, it seems reasonable to regroup the three chambre civile into the same label.

C. Ingesting the dataset

The dataset consists of a corpus of individual judgment written in french.
The files are provided in xml format where the nodes we are interested in are:

```
- <id>
- <formation>
- <contenu>
```

The nested structure of xml is an obstacle for the construction of the dataframe, so we extracted the content of relevant tags into a flat json file which can be easily converted into a csv.

We also dropped duplicate and null content.

The ingested data can be found in /data/interim/output.csv

D. Final dataset

The final dataset dataset has been built in */data/processed/output.csv* and contains 247431 rows and 5 columns. Each row corresponds to a judgement and each column corresponds to a feature, namely :

- case_id (string),
- case_contenu (string),
- case_formation (string),
- case_president (string) and
- case_LABEL (integrer).

III. Building the Features

It follows from the proposed solution that the input and the ouput must have a take a certain form.

A. Building samples

To build the input of our model:

- We tokenized judgment texts into canonical lemmas (no suffix/prefix, no plural form, no capitals) with the keras Tokenizer utility
- Then, we vectorized the lemmas, by turning the top 8 0000 words into a vector.
- Finally, we truncated and padded the input sequences so that they are all in the same length for modeling.

B. Building labels

To convert the labels into a numpy array, we used the keras *to_categorical()* utility that converts a vector of integers into a binary class matrix.

IV. Deep Learning Model

We split the train.csv data set into a training set and a developer test set, with the first 70% of the data being the training set and the latter 30% of the data being the developer test set.

A. Building the Network Layers

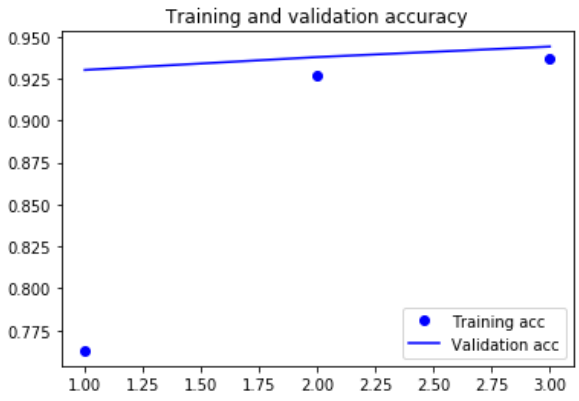
- The first layer is the Embedded layer that uses 128 length vectors to represent each word.

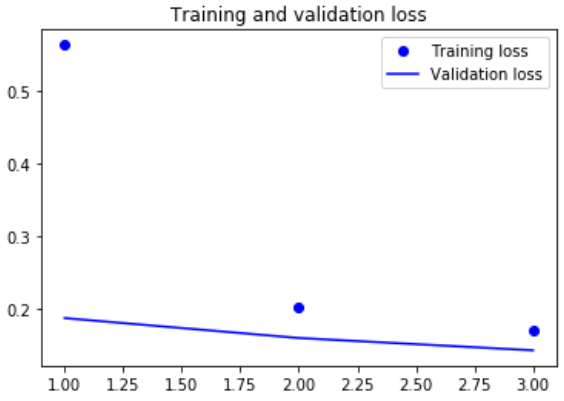
- We had a serious problem of overfitting, so we added a Dropout layer between the Embedding and LSTM layers and the LSTM and Dense output layers. We used the popular (in NLP models) SpatialDropout1D.
- The next layer is the LSTM layer with 64 memory units.
- Finally, because this is a classification problem we uses a Dense output layer with a single neuron and a softmax activation function because it is a multi-class classification.

B. Choosing the Hyperparameters

We followed best practices when we chose the hyperparameters. Because it is a binary classification problem, log loss is used as the loss function (**binary_crossentropy** in Keras). We also used the efficient ADAM optimization algorithm for gradient descent. We limited ourselves to only two epochs because our model overfitted really fast.

C. Loss and accuracy





V. Results

A. Classification Metrics

In accordance with our proposed solution the F1 scores, precision, and recall of our model. Given our multi-label setting, we computed the overall performance with the micro- and macro-averaged of the previously cited metrics.

	Precision	Recall	f1-score	Total

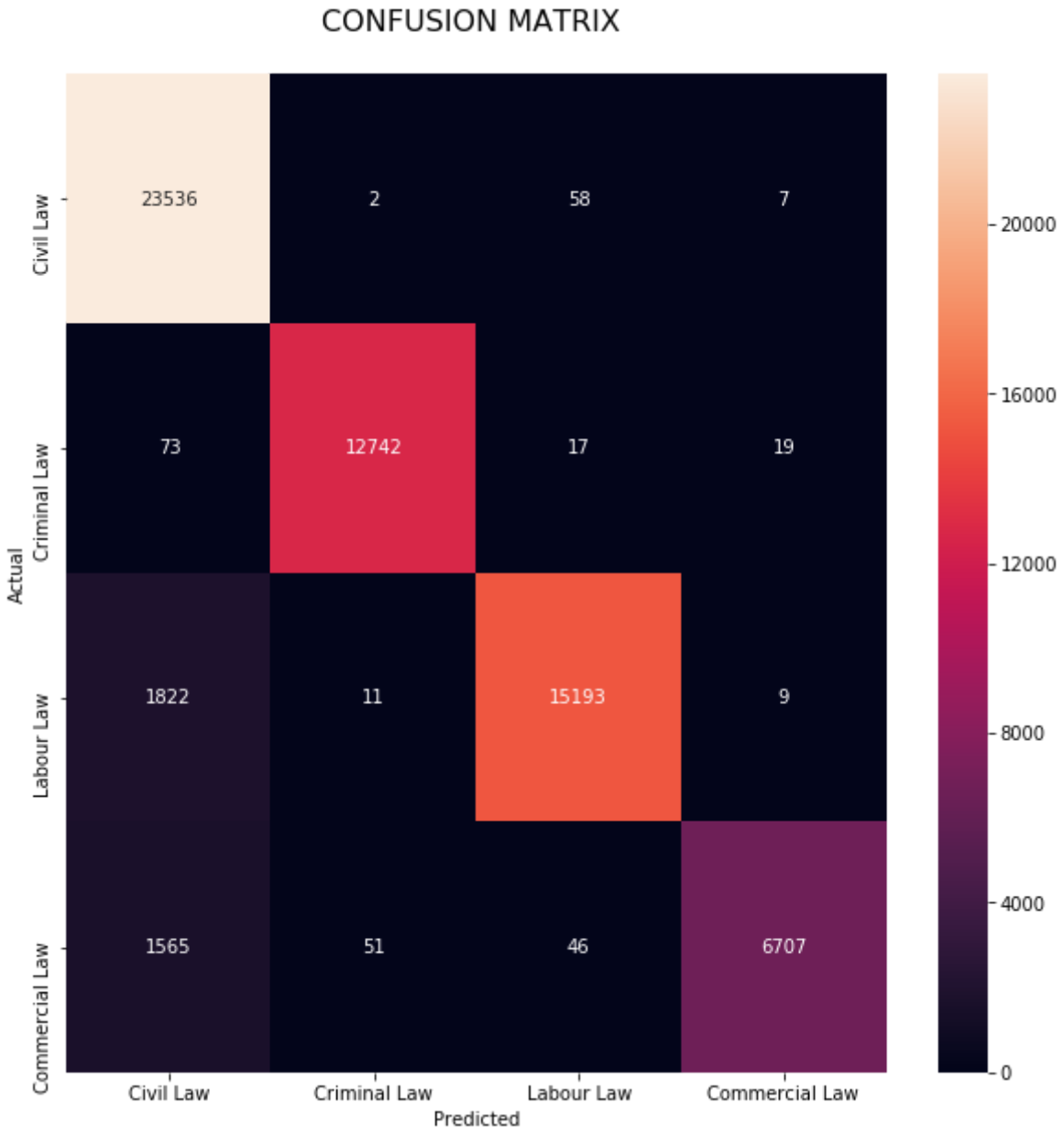
Civil Law	0.88	1.00	0.94	23732
Criminal Law	1.00	0.99	0.99	12762
Labour Law	0.98	0.91	0.94	17025
Comm. Law	1.00	0.79	0.88	8339
micro avg	0.94	0.94	0.94	61858
macro avg	0.96	0.92	0.94	61858
weighted avg	0.95	0.94	0.94	61858

The less than best performance of Commercial Law may be explained by the imbalance between the samples of the classes.

B. Confusion matrix

To improve the performance of our model, we may want to know where it failed. In that order, a confusion matrix will be used.

A confusion matrix may be interpreted as a graph representing where our model misclassified a case.The x axis represents the true class of a case, whereas the y axis represents the predicted class. Thus, the colored diagonal represents where our model predicted the actual class of a case.



We shall take note of two things.

First, the samples of our classes are largely imbalanced towards civil (aprox 25 000 case) and criminal law (aprox 15 000 case).

Secondly, actual Civil cases seems to be often correctly classified whereas commercial and labour case seems to be often misclassified into a civil case.

VI. Discussion

In regards to performance, a simple measure could improve our accuracy. It seems that a lot of misclassification were caused by a bad sampling where some classes vastly outnumbered others, which have been previously noted before.

In a more philosophical tone, I want to question one of our assumptions. We assumed that it is possible to classify cases based on the written transcription of a judgment. I expect this assumption to be fundamentally flawed. Judgments may not state a relevant attribute to classify this case into a certain Legal Area as judges need only to discuss issues relevant to how the case should be resolved. **Thus, we may have proven an effective method to classify (written) judgments but not (material) cases.**

VII. Conclusions

With this project, We learned :

- That imbalanced data tends to over-predict the over-represented class
- How to develop a simple single layer LSTM model
- How to reduce overfitting.

VIII. References

https://machinelearningmastery.com/sequence-classification-lstm-recurrent-neural-networks-python-keras/

Legal Area Classification: A Comparative Study of Text Classifiers on Singapore Supreme Court Judgments Jerrold Soh Tsin Howe*, Lim How Khang , and Ian Ernst Chai**