NLP applied to judicial decision parsing (Predilex Challenge)

Ariane ALIX
Master MVA École Normale Supérieure Paris-Saclay

Summary

- Introduction
- Gender prediction
- Accident/consolidation date prediction
- Final results

Introduction

- Context: extraction of data from "jurisprudences" of trials between a victim and its insurer. (Public summaries of french trials containing the context of the accident, the medical statuses, the financial compensations...)

- Dataset: 1027 raw texts of judicial decisions.
  770 texts used for training/257 texts for testing (ranking the challenge).

- Objectives: extract automatically the most relevant data of the texts: the gender of the victim, the date of the accident and the date of consolidation of their injuries.

Gender prediction

- Features
- Classifier
- Results

Gender prediction - Features

- **Build features**: vectorize the text using a TF-IDF method
- **Selection of key N-grams**: compute X^2 stat between features and ground truth
- **Input table for the classifier**: proportions for each text of the occurrences of the female vs male version of the N-gram (proportion of "née" vs "né"...)
- **Missing values**: replacement by the average of the proportion in female texts and the proportion in male texts to avoid bias.

Gender prediction - Features

- **Build features**: vectorize the text using a TF-IDF method (Bag of Words)
- **Selection of key N-grams**: compute X^2 stat between features and ground truth
- **Input table for the classifier**: proportions for each text of the occurrences of the female vs male version of the N-gram (proportion of "née" vs "né"...)
- **Missing values**: replacement by the average of the proportion in female texts and the proportion in male texts to avoid bias.

	elle/il	née/né	madame/monsieur	subi par madame/monsieur	madame/monsieur a été victime	victime madame/monsieur	verser à madame/monsieur
0	0.296875	0.166667	0.312500	0.000000	0.000000	0.53125	0.500000
1	0.180124	0.333333	0.121622	0.000000	0.476786	0.00000	0.483493
2	0.527027	0.222222	0.842105	0.525568	0.476786	0.53125	0.483493
3	0.306667	0.150000	0.333333	0.525568	0.476786	0.53125	0.483493

Gender prediction - Classifier

- Support Vector Machine (SVM) with a Stochastic Gradient Descent (SGD)

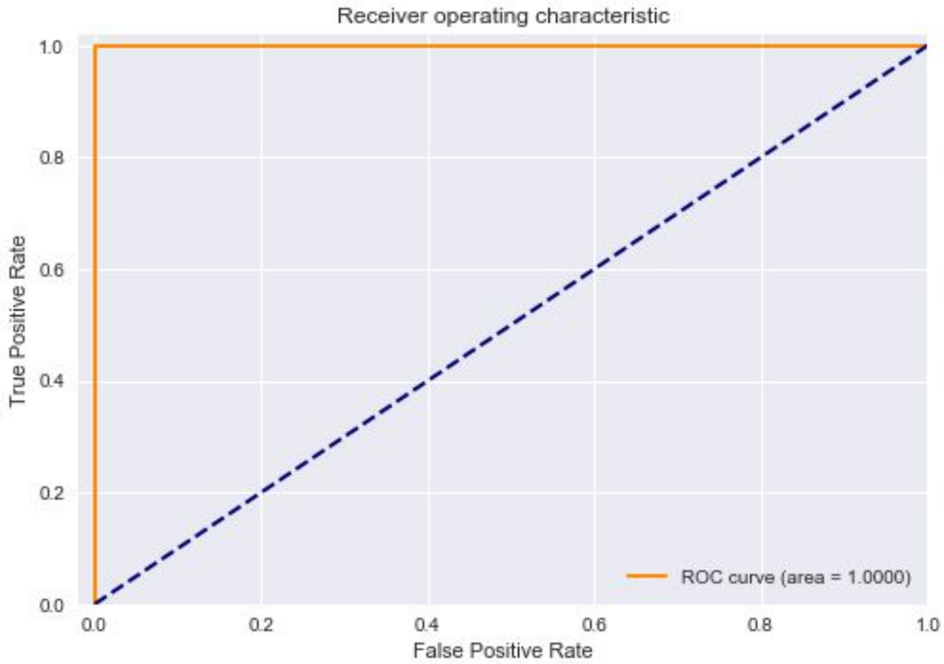
- Cross-validation with 5 folds + GridSearch to look for the parameters giving the highest accuracy

The best parameters found were:

- Hinge loss: assuming the true labels in y are encoded with +1 and -1 and w is the predicted score, the loss is $L_{\text{Hinge}}(y, w) = \max\{1 - wy, 0\}$
- L_1 penalty: regularization in norm L_1 with a 10^{-4} regularization coefficient,
- less than 1000 iterations

Gender prediction - Results

Average accuracy of the cross-validation on training set: 99.74%



Receiver Operating Characteristic curve (female vs all)

Accident and consolidation date prediction

- Extraction of the dates
- Classifier of sentences method 1 (SVM)
- Classifier of sentences method 2 (LSTM)
- Final prediction

Date prediction - 3 parts

We separate the problem in three sub-problems:

- **Part 1** : Extract all the dates of the texts,
- **Part 2** : Identify strings that contain an accident date,
- **Part 3** : choose the most probable accident date given the previous data.

Date prediction - Part 1: **Extraction of the dates**

- Regular expression (Regex) matching methods. Example of patterns: ’*nn/nn/nnnn*’ or ’*nn llllll nn*’ where the *n* are numbers and *l* letters.
- Searching sentence by sentence.
- 27997 dates found in the 770 texts of the training corpus.
- 7 accident dates and 14 consolidation dates not retrieved, (when not n.c or n.a.).
- Storing the subsentences (size 300 characters centered around the date) from which we extracted the dates. We removed the date and replaced it by the word ’date’ to avoid a bias.

Date prediction - Part 2: **Classify subsentences**

- **Input**: subsentences from which we extracted a date in the previous part
- **Output**: probability of the string to contain the accident date

Two methods tried for that part:

- SVM
- Neural Network with an Long Short-Term Memory RNN

Date prediction - Part 2: **SVM method**

Features:

- An SVM algorithm cannot take raw text as input nor perform vectorization itself.
- We vectorize with TF-IDF before feeding the obtained numerical features into the SVM.

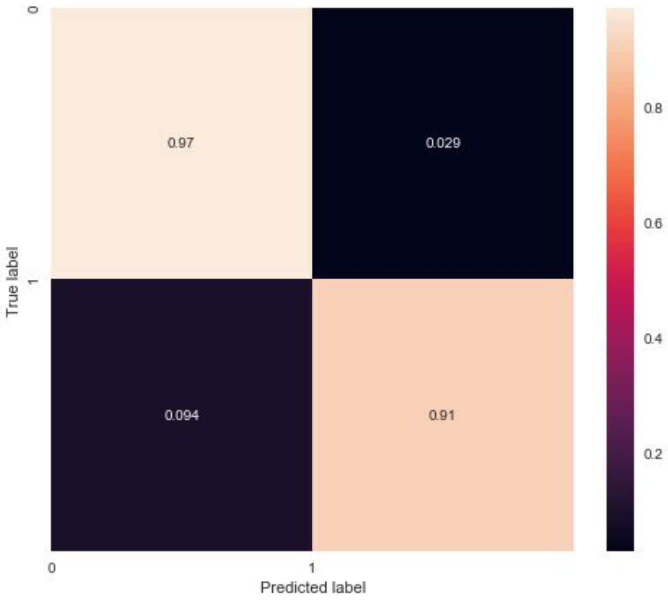
Classifier:

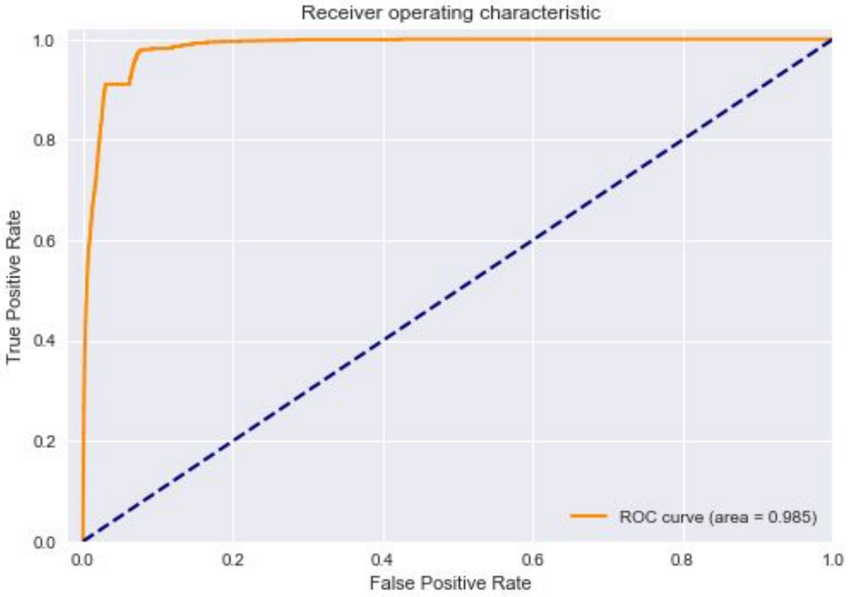
- Support Vector Machine (SVM) with a Stochastic Gradient Descent (SGD)
- Pipeline object to optimize the TF-IDF vectorizer and SVM parameters jointly
- Cross-validation with 5 folds + GridSearch to look for the parameters giving the highest accuracy

Date prediction - Part 2: **SVM method**

Results for the accident date:

- Average accuracy of the cross-validation on training set: 96.52%

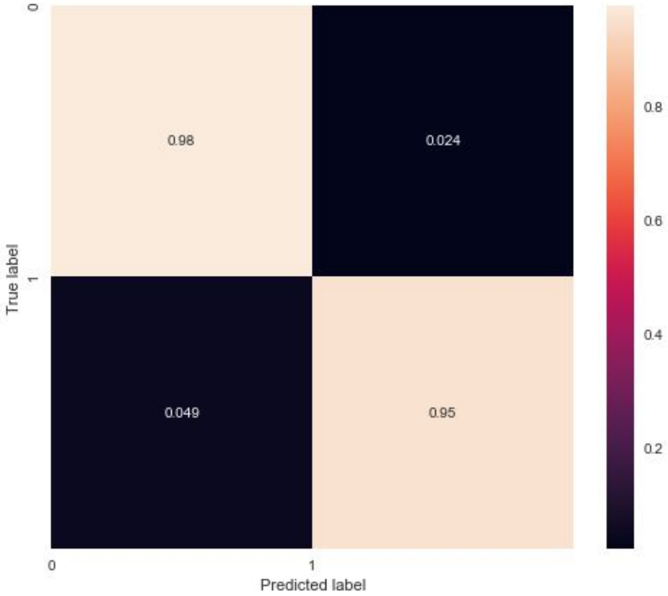


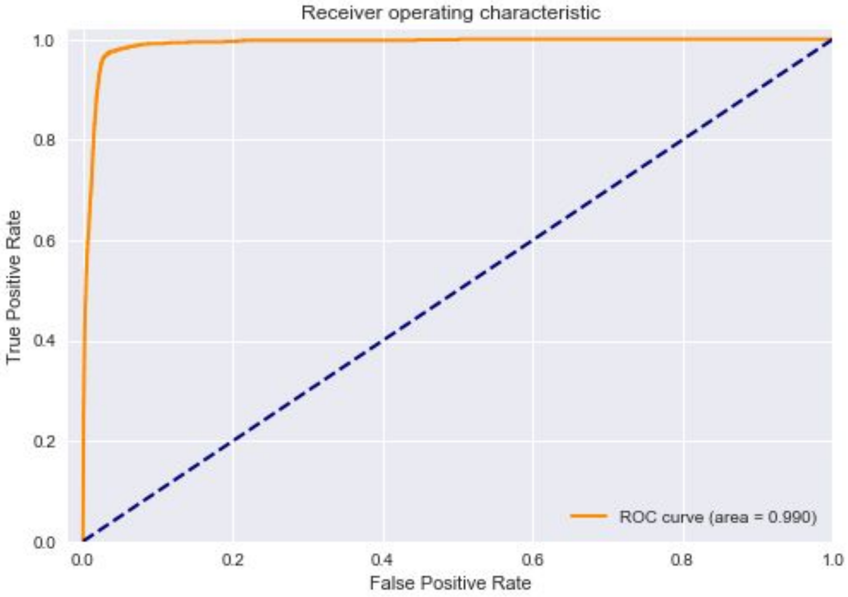


Date prediction - Part 2: **SVM method**

Results for the consolidation date:

- Average accuracy of the cross-validation on training set: 97.50%





Date prediction - Part 2: **Neural Network method**

Classifier:

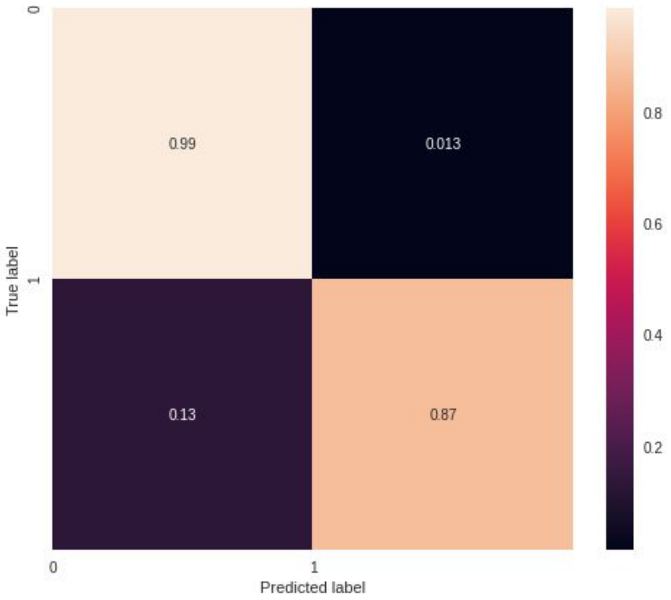
- Computed in the model and optimized with the rest.
- LSTM (Recurrent Neural Network): keeps the order of the words and uses previous words as context to gain a better understanding

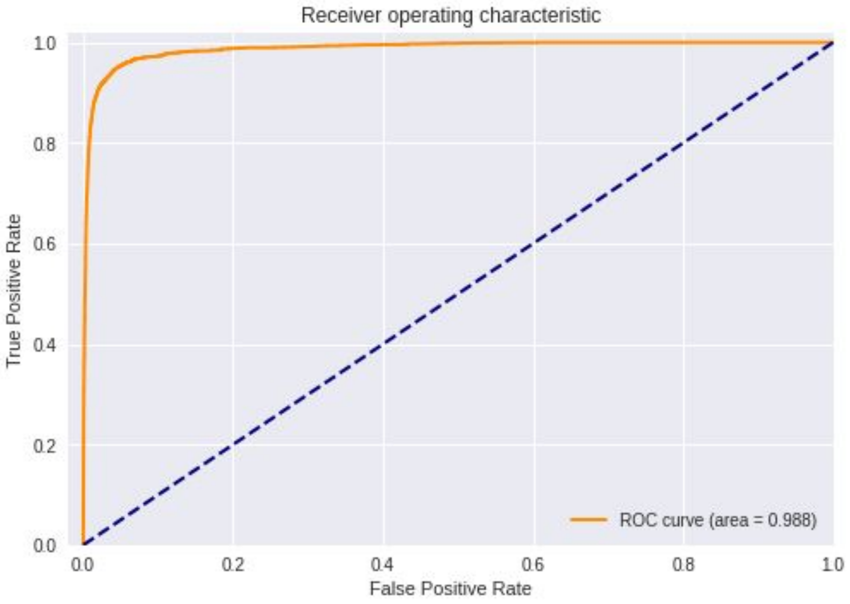
```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, None, 64)          3200000
_________________________________________________________________
lstm (LSTM)                  (None, 64)                33024
_________________________________________________________________
dense (Dense)                (None, 1)                 65
=================================================================
Total params: 3,233,089
Trainable params: 3,233,089
Non-trainable params: 0
_________________________________________________________________
```

Date prediction - Part 2: Neural Network method

Results for the accident date:

- Accuracy: 97.40% on training set, 96.77% on validation set

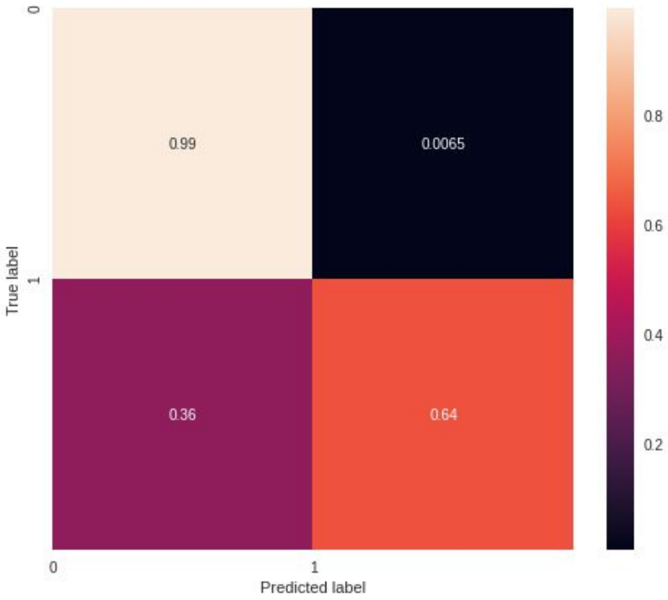


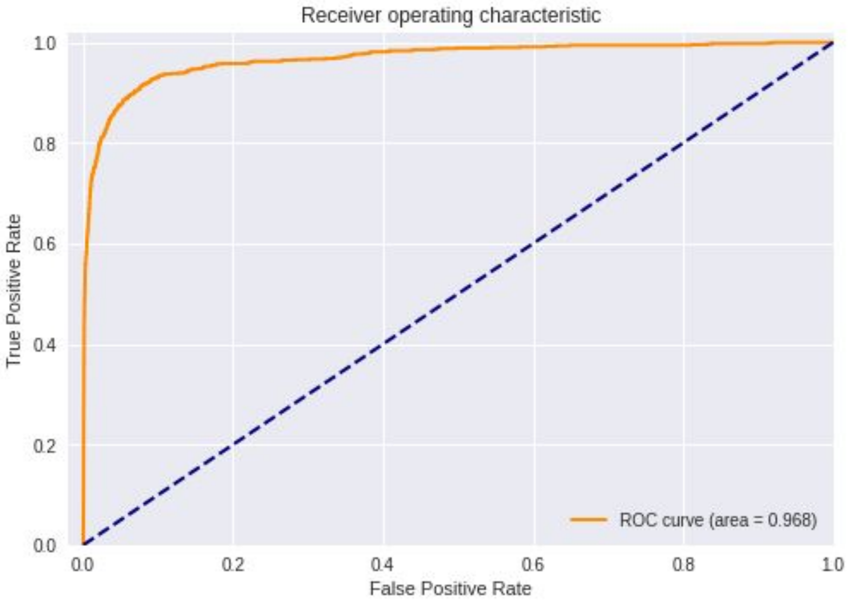


Date prediction - Part 2: Neural Network method

Results for the consolidation date:

- Accuracy: 98.06% on training set, 97.93% on validation set





Date prediction - Part 3: **Prediction of the dates**

- For each subsentence with a date: apply the models of Part 2 to predict if they contain an accident date.
- For each text and each potential date, we choose the date with the highest sum of probabilities given to the subsentences they were extracted from.
- If no subsentence was predicted as containing the accident date, we return 'n.c.' or 'n.a'.
- Annex SVM (AUC 0.982) classifier that predicts if the victim is dead: consolidation date is 'n.a.'

Final results and discussion

Final results

Average accuracy for gender, date of accident and consolidation on training set:

- 88.01% for the SVM method
- 90.13% for the Neural Network (much more expensive in time and computation)

Average accuracy for gender, date of accident and consolidation on training set:

- 80.21% for the SVM method
- Unknown for the Neural Network (could not be tested due to the Challenge submissions limits)

Potential improvements

For the Neural Network:

- Finetune the parameters
- Replace the LSTM by a bi-LSTM (bidirectional)
- Use a weighted cost for imbalanced classes

In general:

- Improve the ‘n.a.’ prediction
- Add overfitting reduction techniques, like data augmentation