Automatic Dating of Documents

Martin Hallén, Gökçen Nurlu, Liang Jensen, Junxion Wang
martin.hallen@epfl.ch, gokcen.nurlu@epfl.ch, liang.jensen@epfl.ch, junxion.wang@epfl.ch
School of Computer and Communication Sciences, EPFL

Introduction

In our **Digital Humanities** project, we have been given 200 years of digitized articles from newspapers *Journal de Genève* (JDG) and *Gazette de Lausanne* (GDL) and our goal was investigating the methods that estimates the date of given text using the dataset.

N° 108 Mardi 8 Mai 1945

JOURNAL DE GENÈVE

NATIONAL, POLITIQUE ET LITTÉRAIRE
FONDÉ EN 1826

La guerre est finie en Europe

La Suisse et la fin des hostilités

Ce qu'il reste à faire

CE SOIR, SONNERIE DES CLOCHES

M. Churchill et le général Eisenhower

Le maréchal Montgomery

Figure 1: A sample frontpage from JDG

Cleaning data

We have observed that there were many articles with unrecoverable OCR errors. Therefore, to work faster with the data, we have converted it to plain CSV files and eliminated the ones with high OCR errors. After that, to improve perfomance of classification methods, we have processed the text further by applying **spell-checking** and **stemming**, which were claimed to increase accuracy [1]. We show in figure 2 that we were able to improve speed and accuracy of classifier by reducing the number of unique words by **10%**.

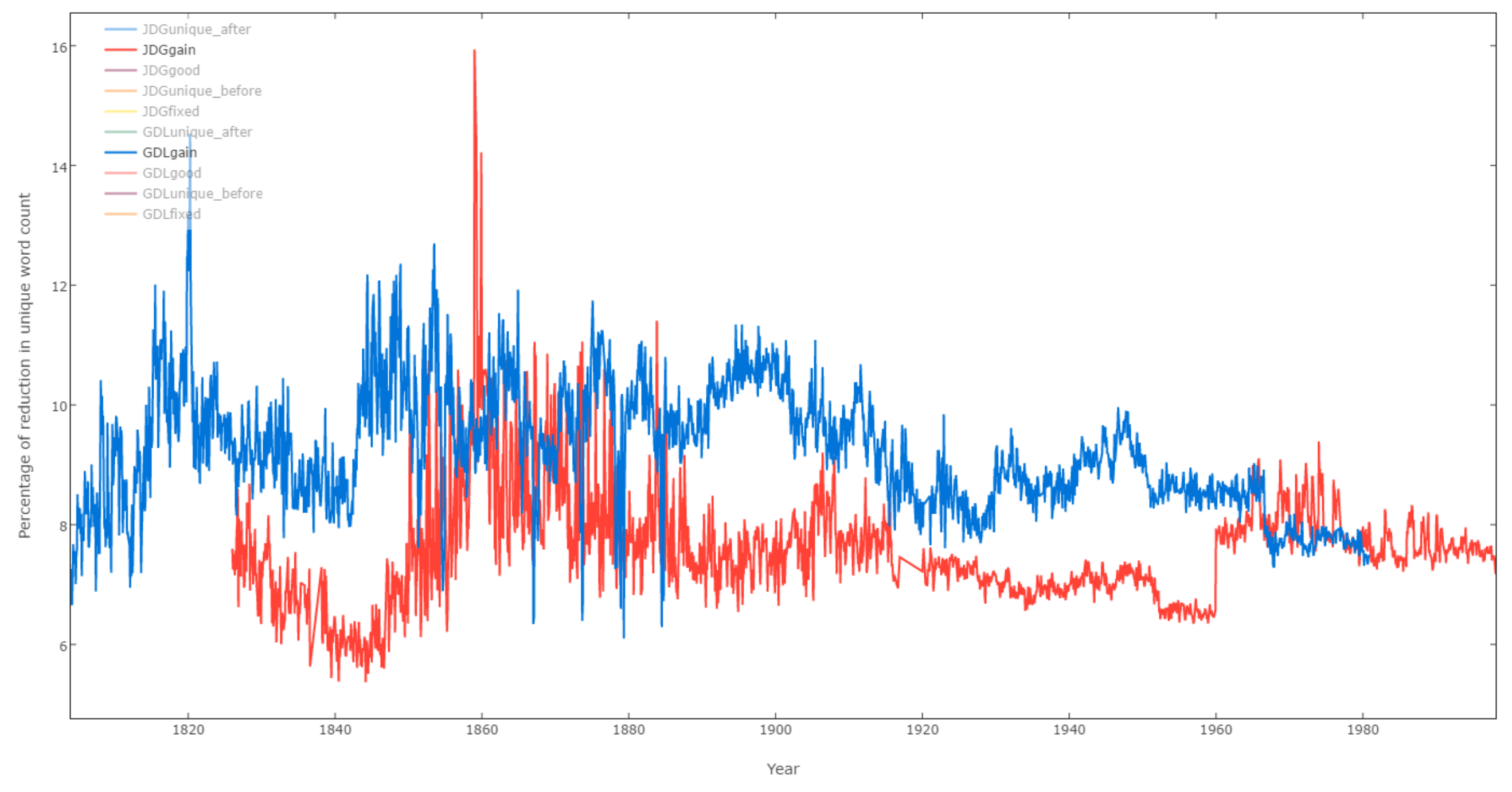

Figure 2: Percentage of reduction in word count per year for both newspapers

Classifying the documents

There are multiple methods of text classification. While some are vast and complex, others use simple probability theory to predict the class of a document.

One of these methods is **Naive Bayes classifier**. While it can be expressed in easy terms, it remains a powerful method for text classification. Given a set of classes $C = \{C_1, C_2, \dots, C_m\}$ and a set of features $x = \{x_1, x_2, \dots, x_n\}$ per document, we want to predict the class given the features, $P(C_k|x_1, x_2, \dots, x_n)$. Naive Bayes classifier presents a simple framework to do this, and it is recognized as a simple yet effective method. Our implementation uses the vocabulary as the features. We see how many times a certain word is used in a document, and calculate the probabilities for the document to be written in a certain year. For more information about the method, we recommend reading about the Naive Bayes classifier. We have divided the data into **train** and **test** data during the project.

Methods we have tried without success

A research project might include attempts that ends without discovering anything. This is true for our project as well.

- Dimensionality reduction: One problem in machine learning is called the *curse of dimensionality*. Our feature set consisted of the vocabulary of the whole newspaper. In total, the vocabulary consisted of more than a million different words. We have tried multiple methods to deal with this, including *principle component analysis* and *singular value decomposition*. While the classification ran faster, the results got worse.
- Regression: Since the classes in our problem represent a continuous variable: year, it would be natural to represent this as a regression problem. We would assume some correlation between a year and the following or preceding year. We have used SVR to perform regression on the data, but did not get any sufficient results.

Results

We have found a clear correlation between the written year of an article and the label we got through classification. We got an accuracy of **33%** on the data set from JDG, where we look at documents for every 5 years. As we can see from figure 3, the earlier years was difficult to classify, and therefore lower this result. Some years achieved an accuracy of more than 60%.

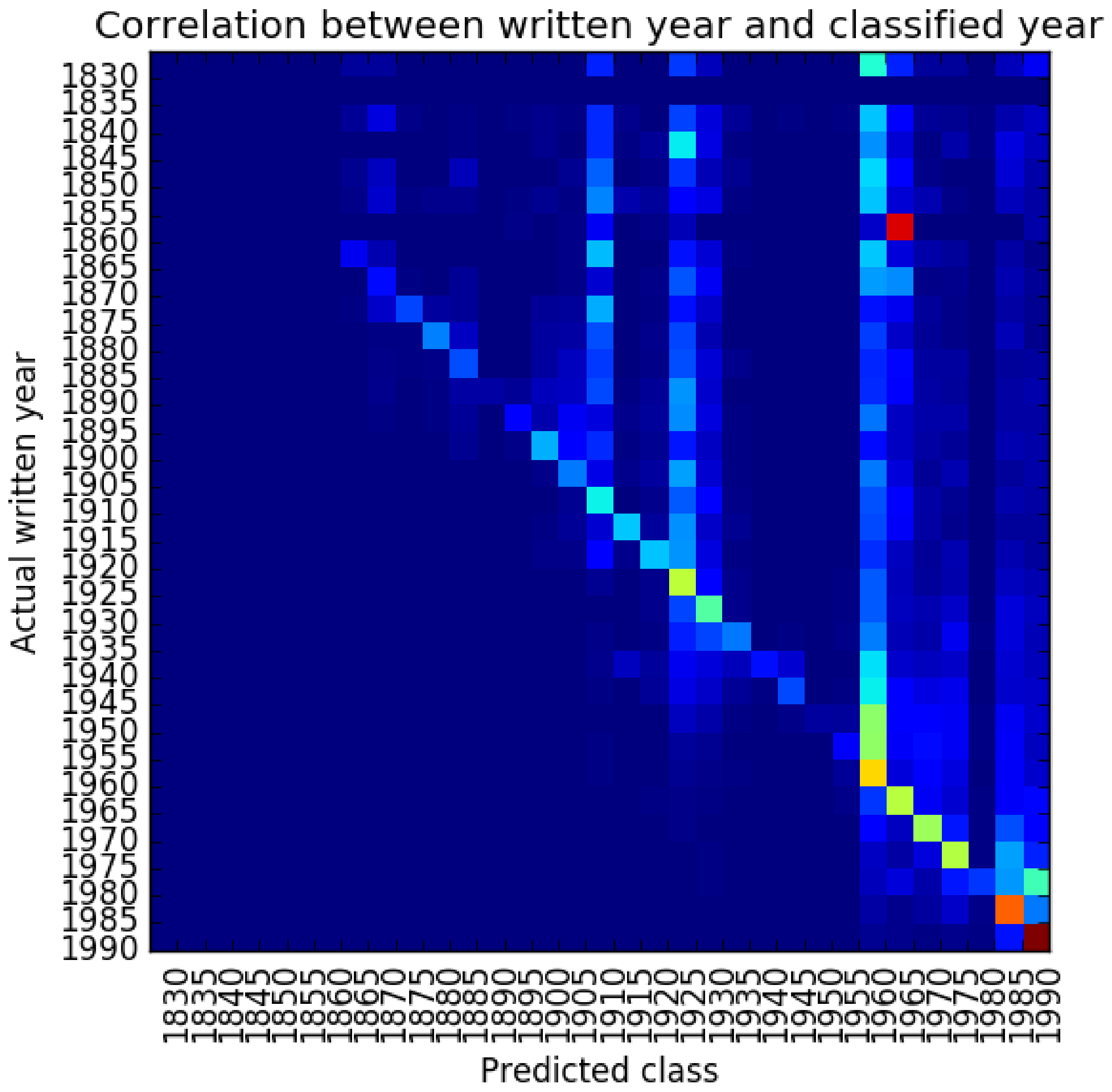

Figure 3: Heatmap of the predicted year of each document against it's true written year.

We have observed a slight skew in the classified result towards later years. We believe this is due to the bad quality of data from the earlier years. It is also interesting to see that some years are more likely to be get documents classified to it. We are happy with the results, but we know that there are improvements. Regression is a possible way to go. We also saw that the amount of data matters a lot in this model. Using more powerful computers with cleaner data can enable the continuation of this project to get improved results.

References

[1] M Ikonomakis, S Kotsiantis, and V Tampakas.
Text classification using machine learning techniques.
WSEAS Transactions on Computers, 4(8):966–974, 2005.

[2] Wen-Tau Yih and Christopher Meek.
Improving similarity measures for short segments of text.
2007.

[3] Introduction to information retrieval.