An automatic partitioning of Gutenberg.org texts

Davide Picca & Cyrille Gay-Croisier
University of Lausanne
[davide.picca,cyrille.gaycroisier]@unil.ch

UNIL | Université de Lausanne
Section des sciences du langage et de l'information

Abstract

Over the last 10 years, the automatic partitioning of texts has raised the interest of the community. The automatic identification of parts of texts can provide a faster and easier access to textual analysis. We introduce here an exploratory work for multi-part book identification. In an early attempt, we focus on *Gutenberg.org* which is one of the projects that has received the largest public support in recent years. The purpose of this article is to present a preliminary system that automatically classifies parts of texts into 35 semantic categories. An accuracy of more than 93% on the test set was achieved. We are planning to extend this effort to other repositories in the future.

Introduction

Over the last 10 years, the automatic partitioning of texts has raised the interest of the community.The need for this type of research is also driven by the compelling use of computational methods for literary texts that often do not meet formatting standards [1]. Nonetheless, there is a twofold difficulty in this field: on the one hand, the heterogeneity of the encoding methods, which do not adhere to a general standard, and, on the other hand, the diversity of literary repositories making it more complex to provide a general method that fits any repository. In a first attempt to address the problem, we focus on *Gutenberg.org* which is one of the projects that has received the largest public support in recent years

Main Objective: automatically classifies parts of Gutenberg.org texts

The purpose of this article is to present a preliminary system that automatically classifies parts of text into 35 semantic categories, listed in Table 1. We are planning to extend this effort to other repositories in the future.

Gutenberg header/footer	Book header/footer	Section	Editorial	Text	Layout
footer	author	chapter number	caption	date	layout
footer license	bibliography	chapter title	character	direct speech	list
footer start	book info	part number	editorial	paragraph	table
header	book title	part title	footnote	place	
header end	epigraph	section number	note	place and date	
header info	glossary	subtitle	play info	quote	
	index				
	table of contents				

Table 1: Parts of text identified in the corpus, sorted by category.

Features Engineering

A selection of 17 features was used. In order to assess their importance, some of them were manually chosen based on observations during the corpus annotation. The 17 features can be split into three different groups: *textual features, boolean features and numerical features* as listed here under.

Textual features:

1. *TFIDF*: This is the raw text processed using TFIDF method. This is the most common feature used in NLP tasks.
2. *First characters*: This feature returns the first five characters of a text, including spaces. This feature seems to be very useful for identifying *titles* and *paragraph*.
3. *Last characters* : This feature returns the last five characters of a text, including spaces. As the previous one, this feature seems to be very useful for identifying *titles* and *paragraph*.
4. *Class of next part*: This feature returns the target class of the next part of text in the document. Most of the time there exists a repetitive pattern in the classes' sequence.
5. *Class of previous part*: This feature returns the target class of the previous part of text in the document.

Boolean features:

1. *Ends with punctuation*: This feature returns True if the last character of the text part is a punctuation mark. The parts *paragraph*, *direct speech* and *quote* often end with a punctuation mark.
2. *First word in capital letters*: This feature returns True if the first word of the text part is in capital letters. The parts *chapter number*, *part number*, *header end*, *footer end* and *book title* often have their first word in capital letters.
3. *Has asterisk* : This feature returns True if there is an asterisk in the text part. The parts *header end*, *footer start* and *layout* often have at least an asterisk.
4. *Has bracket*: This feature returns True if there is one bracket in the text part. The parts *footnote*, *note* and *caption* often have at least one bracket.
5. *Has quote* : This feature returns True if there is one quotation mark in the text part. The parts *direct speech* and *quote* have at least one quotation mark.
6. *Has reporting verbs*: This feature returns True if there is one reporting verb in the text part. Reporting verbs are verbs transmitting the action of speaking, such as "say", "explain" or "think". The part *direct speech* often has one reporting verb.

Numerical features:

1. *Part length*: This feature returns the length of the text part as an integer. The parts *paragraph* are often longer than other parts.
2. *Ratio symbol*: This feature returns the ratio in the text part between the number of symbols and the total number of characters. Symbols are for example currency symbols and hashtags.
3. *Ratio uppercase* : This feature returns the ratio in the text part between the number of uppercase letters and the total number of letters. The parts *header info*, *book title*, *chapter title* and *part title* often have words in uppercase letters.
4. *Ratio word/lemma*: This feature returns the ratio between the number of lemmas and the number of words.
5. *Ratio word with first capital letter*: This feature returns the ratio in the text part between the number of words with their first letter in uppercase and the total number of words. In English, words in titles begin usually with a capital letter. Therefore, the parts *header info*, *book title*, *chapter title* and *part title* often have words with their first letter in uppercase.
6. *Relative position* : This feature returns the relative position of the text part in the document.

Experiment and Results

We approached the problem as a multi-class classification task. The 102,461 target classes of text found in the 111 texts of the corpus were randomly assigned into a training and a test set, given a ratio of 0.33.

The only classes for which the classifiers do not perform well are *dates* and *places*, likely due to the paucity of examples in the training set.

	LinearSVC	KNeighbors	DecisionTree	BernoulliNB
Textual features	**0.940526**	0.876941	0.865288	0.59276
Boolean features	0.584775	0.621152	**0.647887**	0.611008
Numerical features	0.42253	0.721438	**0.765327**	0.483542
All features	**0.953125**	0.946322	0.933369	0.657085

Table 2: F-Measure for each classifier based only on one-group feature and all-combined features

Table 2 shows the F-Measure scores of the three feature groups and all combined features respectively. Figure 1 shows the F-Measure report for each target class and for each classifier. The classes *date*, *place*, *place and date* are poorly predicted likely due to the lack of support items. While there is room for improvement, the reliability of currently available NERs mitigates the severity of such a negative result. Looking at Table 2, we notice that not all features work equally well. There is a clear distinction between textual and non-textual features. While textual attributes correctly predict almost 9 times out of 10, Boolean features have an overall accuracy of 63% and numeric features hardly get close to 50% for LinerSVC and BernoulliNB.

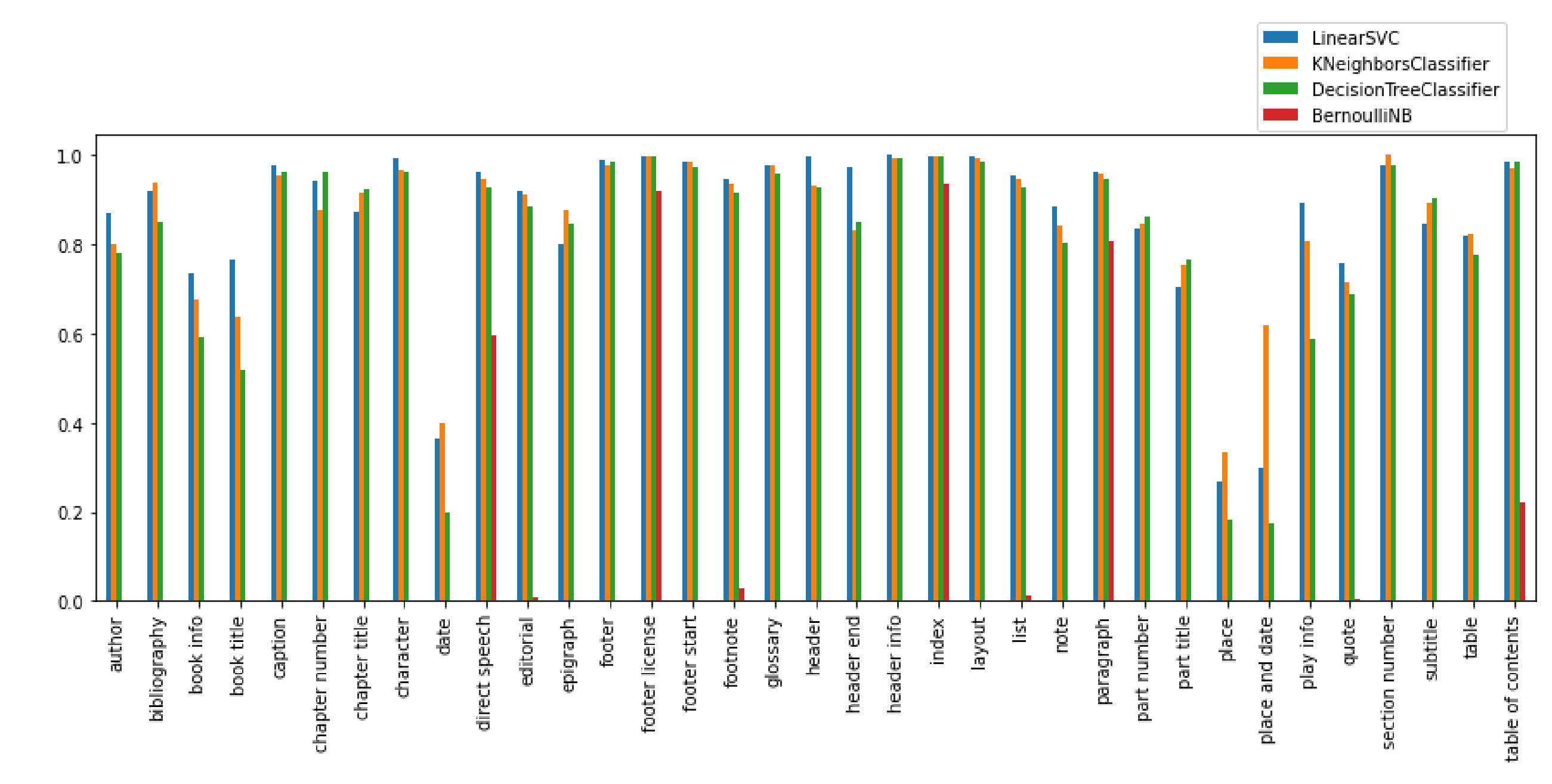

Figure 1: Comparison of F1 Measure on target classes for each classifier

If we analyze the importance of features by group, we clearly notice that the textual features (see Figure 2) achieve an accuracy of more than 75% and can accommodate almost any class reflecting the importance of the spelling and textual features for this task.

In particular, the textual features *Type of the previous part* and *Type of the next part* help to classify the sections according to their location and the surrounding parts in the text. For example, these two features identify the *index* with almost perfect accuracy, while other textual features do not work well for that specific class.

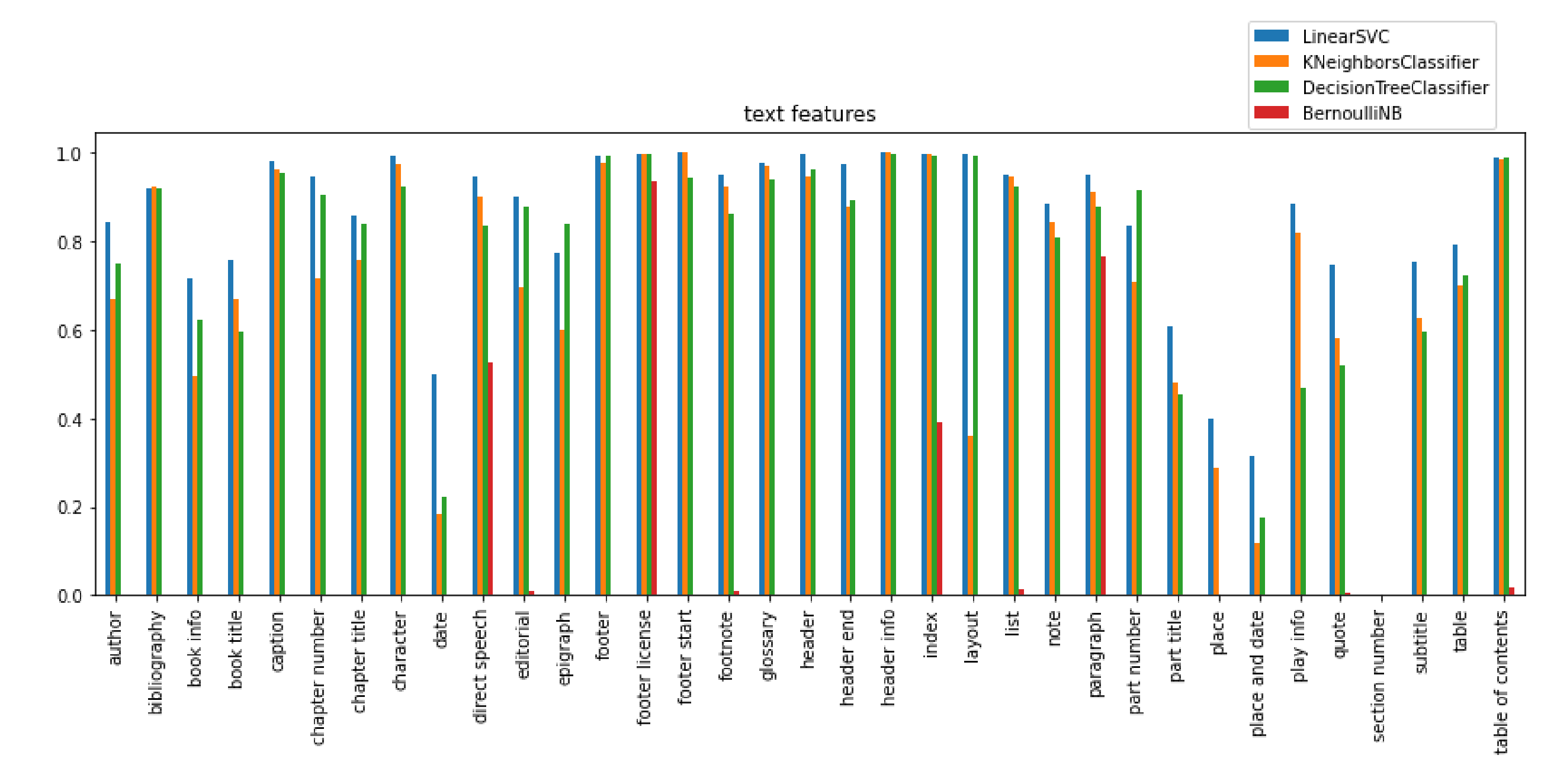

Figure 2: F1 measure for text features

Then, Boolean features do not perform well on the majority of classes. Those features were developed primarily to identify specific parts of the text. Like boolean features, numerical features fail to predict the majority of text parts. Interestingly, as far as numerical features are concerned, they have a different effect depending on the algorithm used. In fact, they seem to perform better with the DecisionTree algorithm than the others. Just as the BernoulliNB algorithm seems to outperform with the Boolean features.

However the general system shows very good results reaching scores above 90% for many classes. In particular, looking more closely at Figure 1, we can observe that some specific parts such as *captions*, *numbers* and *titles* of the chapter, as well as the *direct speeches* and *footers* achieve results above 90%.

Conclusion

This paper presents a system for the automatic identification of parts of literary texts in the Gutenberg repository. With an overall accuracy of 93%, the system offers satisfying results. The best performing features are the textual ones, which succeed in predicting almost all classes. Boolean and numerical features did not have a major influence on the classification, but help to identify specific parts of text. The two most recurrent classes, *direct speech* and *paragraph*, have been identified with a degree of precision of 95%. This high precision score is an encouraging result, as these two classes are the most relevant parts for textual analysis in literature.

References

[1] Mattia Egloff and Davide Picca. The Project Gutenberg Ontology. In *European Association for Digital Humanities (EADH)*, Galway, Ireland, 2018.

[2] Mattia Egloff, Davide Picca, and Alessandro Adamou. Extraction of character profiles from the gutenberg archive. In Emmanouel Garoufallou, Francesca Fallucchi, and Ernesto William De Luca, editors, *Metadata and Semantic Research*, pages 367–372, Cham, 2019. Springer International Publishing.