


An Automatic Partitioning of Gutenberg.org Texts

Davide Picca   

University of Lausanne, Switzerland

Cyrille Gay-Crosier 

University of Lausanne, Switzerland

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.

2012 ACM Subject Classification Computing methodologies; Computing methodologies → Language resources

Keywords and phrases Digital Humanities, Machine Learning, Corpora

Digital Object Identifier 10.4230/OASICS.LDK.2021.35

1 Introduction

Over the last 10 years, the automatic partitioning of texts has raised the interest of the community [6]. In fact, while humans perform text segmentation smoothly during reading, automatic approaches struggle with the problem of inferring the paragraphemic uses of signs. 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 [13, 3, 8, 14]. In fact, such an identification would make a finer textual analysis possible, based on the narrative parts of the text (i.e., direct speech, footnote, etc.). 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 order to tackle this issue, we introduce here an exploratory work for multi-part book identification. In a first attempt to address the problem, we focus on *Gutenberg.org*¹ which is one of the projects that has received the largest public support in recent years [9, 21, 16]. 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.

2 Related Work

The tracks proposed by the INEX and ICDAR book structure extraction competitions [6, 15, 7] share with our paper the same general topic. In these tracks, participants are asked to submit automated methods for more accurate identification of text parts such as *Abstract*, *Introduction*, *Methods*, *References*. Nonetheless, with respect to these challenges, our work aims to use a manually pre-defined set of categories, which is more related to the work

¹ <https://www.gutenberg.org/>



proposed by [18]. Our article differs in two main respects: on the one hand, we introduce a finer definition of the structural categories extending them from 10 to 35 and, on the other hand, we focus on classifying the parts of the text rather than classifying the pages themselves. Other authors such as [10, 5, 22] also introduce works whose systems rely on parsing the table of contents rather than relying on the content of the book itself. Most of the contributions analyzing the textual content is related to phrases and paragraphs segmentation [20, 2, 17]. Although the task has a relatively solid tradition, it focuses on identifying a specific part of the book's content without taking into account the 35 categories as shown in Table 1. The choice to consider a broader spectrum of categories has a twofold reason. On the one hand, the frequency of each of these categories (See Figure 1) justifies the interest of counting them as relevant. In this way, we also assume that we cover a sufficiently large number of possibilities should these categories be expanded to include other repositories of literary texts. On the other hand, the choice is motivated by the fact that some minor categories (such as epigraphs or figure captions) play a major role in the study of certain literary and linguistic phenomena. Indeed, a great deal of information relevant to scholars working in the literary field resides in very fine-grained categories. By having introduced some subcategories to the macro-categories defined in the Table 1, even though they are widely less used, we believe we are encouraging scholars in such fields to use this tool for their research.

3 Experiment

3.1 Dataset construction

For the experiment, we rely on the Gutenberg Project repository since it is one of the most used repositories in the Digital Humanities [11, 12, 4], with a variety and well-balanced composition of texts. In fact, it consists of more than 50,000 eBooks (i.e., raw text files) of many different genres, like fiction, poetry, journal articles or scientific papers. A corpus of 169 texts was randomly collected by Project Gutenberg using the DHTK library provided by [19]. The corpus includes texts from different eras, genres and authors, to avoid any bias. Out of 169 texts, 111 have finally been retained, so as to have only texts in English ². Each text has been downloaded as a *.txt* file.

An initial manual analysis was performed to identify regular patterns to mark the categories. Then, an automatic file segmentation was applied using regular expressions with the intention of capturing the 35 categories. Finally, an annotator checked the entire dataset for double-checking. On the one hand, the annotator checked the accuracy of the algorithm in capturing each category, and on the other hand, it evaluated the recall of the algorithm in order to check that the algorithm did not miss any relevant categories. Since the task was performed by only one annotator, no measure of agreement between annotators was performed, but each part of the texts was labeled according to one of the categories described in the Table 1 until the entire corpus were labeled. A final distribution of the 35 selected categories is shown in Figure 1.

² Dataset and code are freely available here <https://gitlab.com/cgaycro1/gutenberg-files-tagging.git>. A request access can be sent through the gitlab platform

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

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				

3.1.1 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, others were drawn from the McConnaughey’s work [18]. The 17 features can be split into three different groups: *textual features*, *boolean features* and *numerical features* as listed here under.

Textual features:

TFIDF: This is the raw text processed using TFIDF method. This is the most common feature used in NLP tasks.

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*.

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*.

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.

Class of previous part: This feature returns the target class of the previous part of text in the document.

Boolean features:

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.

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.

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.

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.

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.

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:

Part length: This feature returns the length of the text part as an integer. The parts *paragraph* are often longer than other parts.

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.

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.

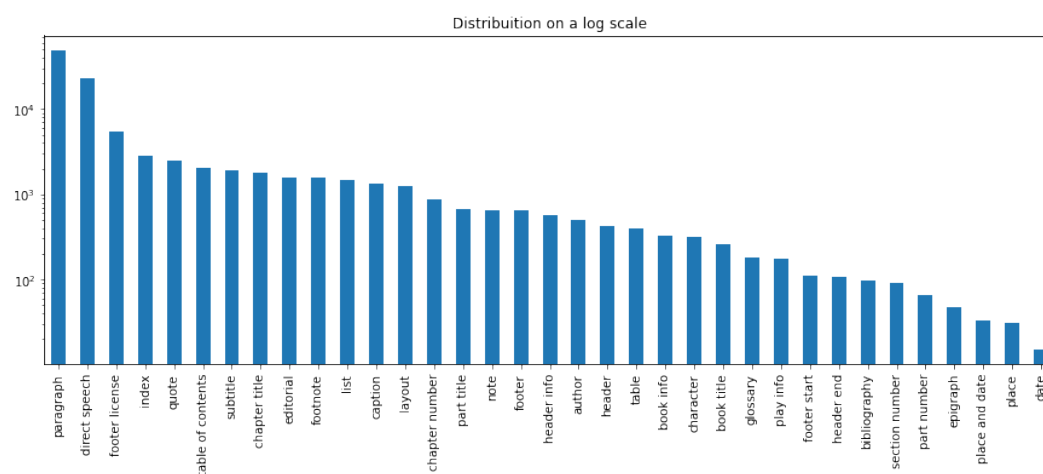
Ratio word/lemma: This feature returns the ratio between the number of lemmas and the number of words.

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.

Relative position : This feature returns the relative position of the text part in the document.

3.2 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 (as described in 3.1) were randomly assigned into a training and a test set, given a ratio of 0.33 with the distribution shown in Figure 1



■ **Figure 1** Distribution of target classes on a log scale.

To explore the problem we compared four inherently multiclass classifiers as suggested by [1] and shown in Table 2. Moreover, in order to offset the class imbalance, where possible, we weighed the classes using the following formula: $\frac{|X|}{|T| \times f(T)}$ where X is the cardinality of samples, T is the total number of target classes and f is a function counting the number of elements $t \in T$ whose values lie in successive integer bins.

Three algorithms out of four achieve an overall accuracy of 93% on the test set as shown in Table 2. It can be noticed that, with the exception of the Bernoulli Naive-Bayes classifier, all other classifiers perform encouragingly for each category, crossing an F-Measure of over 90% for almost every class.

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.

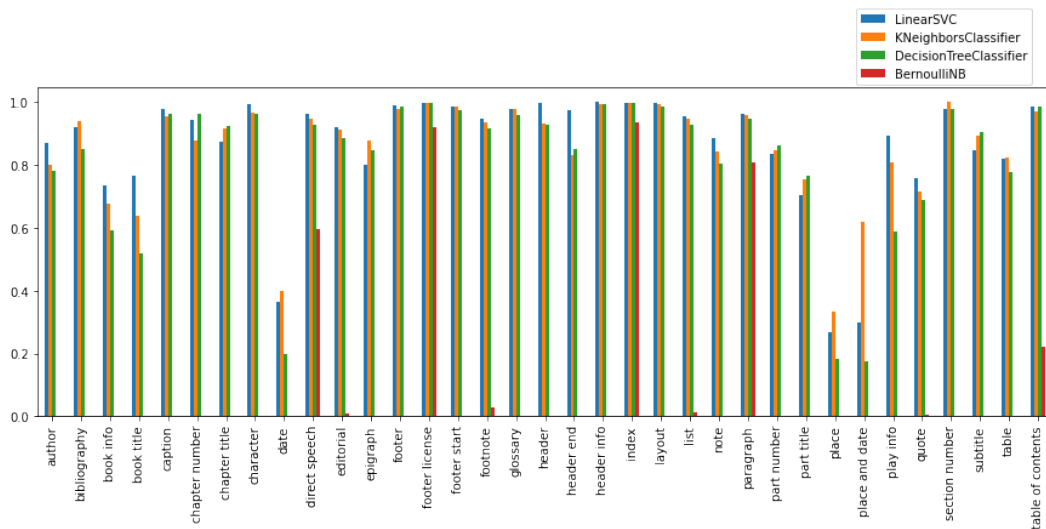
One-feature classifiers and combined-features classifiers were built to compare the performance of individual features on the classification process, similarly as proposed by [20].

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

	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 shows the F-Measure scores of the three feature groups and all combined features respectively.

Figure 2 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 2** 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 3) 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.

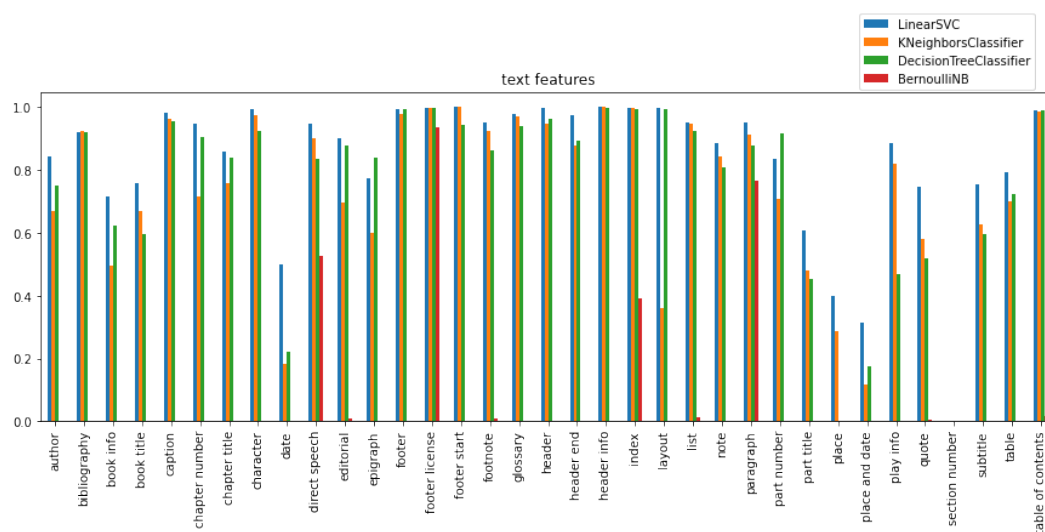


Figure 3 F1 measure for textual features.

Then, Boolean features (see Figure 4) do not perform well on the majority of classes. Those features were developed primarily to identify specific parts of the text.

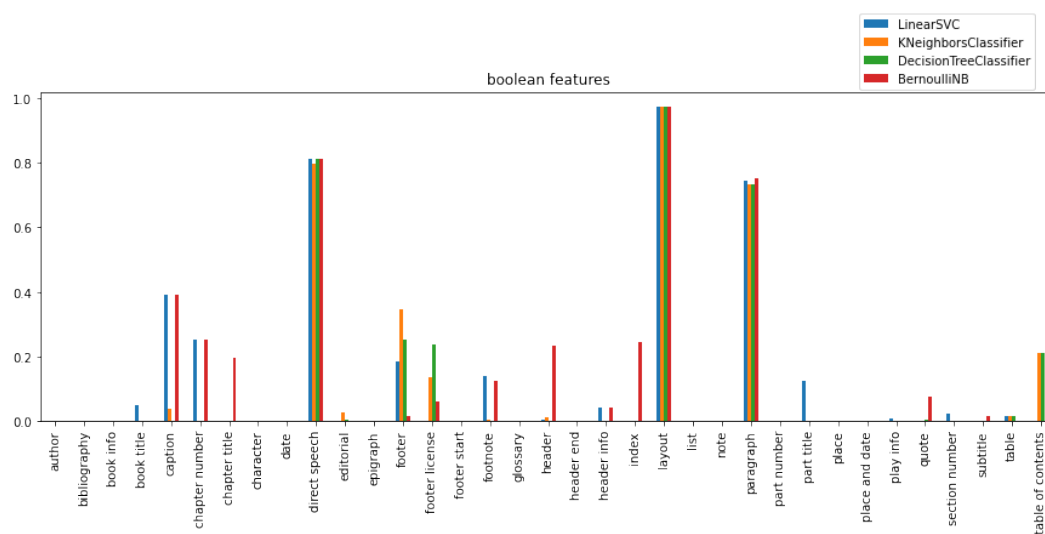


Figure 4 F1 measure for Boolean features.

The feature *Has asterisk* was meant to identify the *layout*, as there are almost always asterisks. According to the table, it predicts a *layout* category with an accuracy of 98%. Similarly, the Boolean feature *Has quote* is effective to identifying *direct speeches* thanks to the presence of quotation marks. Other Boolean features were not able to predict other classes. This is the case for the feature *Has bracket*, which was meant to identify *footnotes* and *captions*, as these parts are almost always contained between brackets in Gutenberg texts.

Like boolean features, numerical features (see Figure 5) 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.

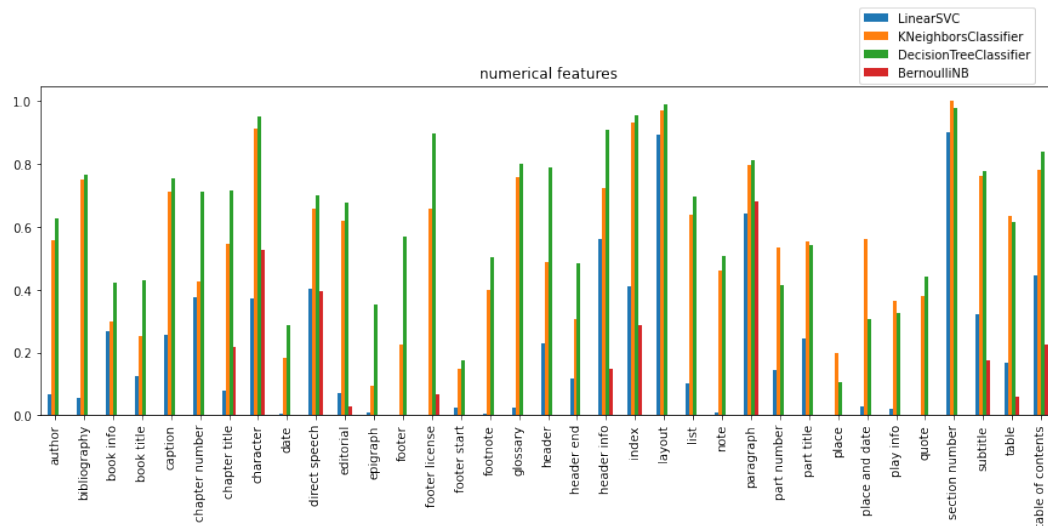


Figure 5 F1 measure for numeric features.

It is interesting to note that parts such as *direct speeches* or *quotes*, which in principle are similar in spelling, achieve results with a high percentage deviation due most likely to the lower number of supports for quotations.

However the general system shows very good results reaching scores above 90% for many classes. In particular, looking more closely at Figure 2, 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%.

4 Conclusion and Future Work

This paper presents a system for the automatic identification of parts of literary texts in the Gutenberg repository. Its aim is to provide scientists in the field of humanities with a tool to ease and fasten the access to textual analysis by identifying the narrative parts that are relevant to the textual analysis. 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.

In the future, further attention will be given to textual features. It would be interesting to explore these results further, by adding new textual features in order to improve the overall classification accuracy. In addition, we are planning to implement a systematic comparison between different classification algorithms. Our aim is to explore thoroughly the influence of each text feature in order to gain a better comprehension of the phenomenon.

References

- 1 Mohamed Aly. Survey on multiclass classification methods. *Neural Netw*, 19:1–9, 2005.
- 2 Ngo Xuan Bach, Nguyen Le Minh, Tran Thi Oanh, and Akira Shimazu. Learning logical structures of paragraphs in legal articles. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 20–28, Chiang Mai, Thailand, 2011. Asian Federation of Natural Language Processing. URL: <https://www.aclweb.org/anthology/I11-1003>.
- 3 Julian Brooke, Adam Hammond, and Graeme Hirst. GutenTag: an NLP-driven Tool for Digital Humanities Research in the Project Gutenberg Corpus. In *Proceedings of the Fourth Workshop on Computational Linguistics for Literature*, pages 42–47, 2015. doi:10.3115/v1/w15-0705.
- 4 R Bucher. *Classification of Fiction Genres: Text classification of fiction texts from Project Gutenberg*. diva-portal.org, 2018.
- 5 Hervé Déjean and Jean Luc Meunier. On tables of contents and how to recognize them. *International Journal on Document Analysis and Recognition*, 2009. doi:10.1007/s10032-009-0078-8.
- 6 Antoine Doucet, Gabriella Kazai, Bodin Dresevic, Aleksandar Uzelac, Bogdan Radakovic, and Nikola Todic. ICDAR 2009 book structure extraction competition. In *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 2009. doi:10.1109/ICDAR.2009.280.
- 7 Antoine Doucet, Gabriella Kazai, and Jean Luc Meunier. ICDAR 2011 book structure extraction competition. In *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 2011. doi:10.1109/ICDAR.2011.298.
- 8 Mattia Egloff and Davide Picca. The Project Gutenberg Ontology. In *European Association for Digital Humanities (EADH)*, Galway, Ireland, 2018.
- 9 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.
- 10 Liangcai Gao, Zhi Tang, Xiaofan Lin, Xin Tao, and Yimin Chu. Analysis of book documents' table of content based on clustering. In *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 2009. doi:10.1109/ICDAR.2009.143.
- 11 M Gerlach and F Font-Clos. A standardized project gutenberg corpus for statistical analysis of natural language and quantitative linguistics. *Entropy*, 2020.
- 12 OL Goodloe. *Applications of Deep Neural Networks to Neurocognitive Poetics: A Quantitative Study of the Project Gutenberg English Poetry Corpus*. repository.asu.edu, 2019.
- 13 Shesen Guo, Ganzhou Zhang, Run Zhai, and Zehua Song. Distribution of English syllables in e-books of Project Gutenberg and the evolution of syllable number in two subcorpora. *Digital Scholarship in the Humanities*, 30(3):344–353, 2015. doi:10.1093/l1c/fqu013.
- 14 Arthur M. Jacobs. The Gutenberg English Poetry Corpus: Exemplary Quantitative Narrative Analyses. *Frontiers in Digital Humanities*, 5, 2018. doi:10.3389/fdigh.2018.00005.
- 15 Gabriella Kazai, Antoine Doucet, Marijn Koolen, and Monica Landoni. Overview of the INEX 2009 book track. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2010. doi:10.1007/978-3-642-14556-8_16.
- 16 Evgeny Kim and Roman Klinger. Who feels what and why? annotation of a literature corpus with semantic roles of emotions. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1345–1359, Santa Fe, New Mexico, USA, 2018. Association for Computational Linguistics. URL: <https://www.aclweb.org/anthology/C18-1114>.
- 17 Jiwei Li, Thang Luong, and Dan Jurafsky. A hierarchical neural autoencoder for paragraphs and documents. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1106–1115, Beijing, China, July 2015. Association for Computational Linguistics. doi:10.3115/v1/P15-1107.

- 18 Lara McConnaughey, Jennifer Dai, and David Bamman. The labeled segmentation of printed books. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 737–747, Copenhagen, Denmark, 2017. Association for Computational Linguistics. doi:10.18653/v1/D17-1077.
- 19 Davide Picca and Mattia Egloff. DHTK: The Digital Humanities ToolKit. In *Workshop on Humanities in the Semantic Web – WHiSe II*, pages 1–6, 2017.
- 20 Caroline Sporleder and Mirella Lapata. Automatic Paragraph Identification: A Study across Languages and Domains. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 72–79, 2004.
- 21 Joseph Worsham and Jugal Kalita. Genre identification and the compositional effect of genre in literature. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1963–1973, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL: <https://www.aclweb.org/anthology/C18-1167>.
- 22 Zhaohui Wu, Prasenjit Mitra, and C. Lee Giles. Table of contents recognition and extraction for heterogeneous book documents. In *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 2013. doi:10.1109/ICDAR.2013.244.