# Daniel@FinTOC-2019 Shared Task : TOC Extraction and Title Detection

**Emmanuel Giguet**
Normandie Univ, UNICAEN,
ENSICAEN, CNRS, GREYC
14000 Caen, France
emmanuel.giguet@unicaen.fr

**Gaël Lejeune**
STIH, EA 4509
Sorbonne University
75006 Paris, France
gael.lejeune@sorbonne-universite.fr

## Abstract

We present different methods for the two tasks of the 2019 FinTOC challenge: Title Detection and Table of Contents Extraction. For the Title Detection task we present different approaches using various features : visual characteristics, punctuation density and character n-grams. Our best approach achieved an official F-measure score of 94.88%, ranking 6 on this task. For the TOC extraction task, we presented a method combining visual characteristics of the document layout. With this method we ranked first on this task with 42.72%.

## 1 Introduction

This paper describe our participation to the FinTOC-2019 Shared Task dedicated to Financial Document Structure Extraction (Rémi Juge, 2019). We submitted results for the two sub tasks: *Title detection*, a binary classification task focusing on detecting titles in financial prospectuses, and *TOC structure extraction* aiming at identifying and organizing the headers of the document according to its hierarchical structure.

Title detection and Table of Content (ToC) extraction are two important tasks for Natural Language Processing and Document Analysis, in particular in the context of digital libraries and scanned books. ToC extraction aims to retrieve or create a ToC in documents where the logical structure is not explicitly marked, difficult to detect or "computationnaly opaque" (de Busser and Moens, 2006). ToC extraction enriches the access to searchable text, in particular in the domain of digital humanities in which the texts are usually longer than in other domains involving Information retrieval (IR) and Natural language Processing (NLP). Rich logical structures is exploited for instance for document classification and clustering (Doucet and Lehtonen, 2007; Ait Elhadj et al., 2012).

Title detection can be a preliminary task for ToC extraction since it will help to detect a page with an existing ToC or it can help to find the bricks to reconstruct the ToC. It can also help classification systems which rely on titles and text structure to detect salient information in textual data(Lejeune et al., 2015). Salient sentences detection can as well be improved via text structure information (Denil et al., 2014).

In section 2 we will give a brief presentation of existing techniques for ToC extraction and title detection tasks. We will present our systems[1] in section 3 and Section 4 will be dedicated to conclusion and perspectives.

## 2 State of the Art

Textual data is often described as "unstructured data" as opposed to structured data like databases or XML data for instance. However, it is probably more accurate to describe textual data as "computationnaly opaque" so that only the file format can be qualified structured, unstructured or semi-structured. The logical structure of natural language data is probably more important for human understanding than the syntactic structure. For instance, in press articles important information is found in the titles and subtitles, making the detection of titles important for improving web indexation (Changuel et al., 2009) or downstream NLP tasks (Huttunen et al., 2011; Daille et al., 2016; Tkaczyk et al., 2018). Regarding title detection task itself, (Xue et al., 2007) showed that for web pages, the size of the characters is not enough to detect titles but (Beel et al., 2013) showed to the contrary that for PDF document it is the best heuristic (70% accuracy).

Visual and textual information can be combined

[1] Code source available online : https://github.com/rundimeco/daniel_fintoc2019

to make a difference between title and non titles, as in boilerplate removal (Lejeune and Zhu, 2018; Alarte et al., 2019).

There are two main types of ToC extraction techniques: those relying on the detection of ToC pages and those relying on the book content. The ICDAR Book Structure Extraction competitions results (Doucet et al., 2013) showed that the most promising systems are hybrid ones, (Nguyen et al., 2017) showed how combining multiple systems can lead to significant improvements in the results. As in boilerplate detection and removal, geometric relations and font information form the main feature types for ToC extraction (Klampfl et al., 2014).

# 3 Methods and Results

## 3.1 TOC Extraction

In order to participate to this first edition and to deliver results in a very short time, we made quite strong assumptions and some shortcomings. Our strategy relies on the detection of the Table of Content (ToC). A simple fallback strategy based on the whole content analysis is used when no ToC pages are detected.

In previous INEX Book Structure Extraction Competitions, we used to consider only the whole document to extract the structure (Giguet and Lucas, 2010a,b; Giguet et al., 2009). Taking into account the whole content of the document has many advantages. First, it allows to handle documents without ToC. Second, it permits to extract titles that are not included in the ToC, such as lower-level titles or preliminary titles. Thus, it reflects the real structure of the document. Third, and not the least, it avoids having to manage or to process erroneous ToCs. Indeed, the ToC of a document may not be synchronized with the actual version of the document when the author forget to update it. It may also contain entries that are not titles, for instance a paragraph incorrectly labelled as a title, or wrong page numbers. Those cases are not rare.

Although these issues are well known and plead in favor of an extraction from the whole content, it is interesting to work with a different approach. Thus we choose to locate ToC pages, to extract their content, and to submit the result as the document structure. Our expectations is to have a good precision but a low recall due to missing or incomplete ToCs.

### 3.1.1 Technical assumptions

The experiment is conducted from PDF documents to ensure the control of the entire process. The document content is extracted using the `pdf2xml` command (Déjean, 2007).

We assume that the PDF reports are automatically generated by the PDF driver of a word processor. Thus, we do not check if the document is a scanned document or if it is the output of an OCR application. Consequently, we do not consider possible trapezoid or parallelogram distortion, page rotation or curved lines. This assumption simplifies the initial stages: baselines are inferred from the coordinates on the x-axis; left, right and centered alignments are inferred from the coordinates on the y-axis.

We also assume that PDF drivers serialize the content of a page area by area, depending on the page layout. A content area corresponds to a page subdivision such as a column, a header, a footer, or a floating table or figure. When a content area is processed, we assume that characters and lines are serialized in reading order, so that there is no ordering problem to consider. Thus, when parsing a page, we expect to find the ToC entries serialized in reading order, and we expect to find the different parts of each ToC entry serialized in reading order.

However, content areas are represented neither in the PDF structure nor in the `pdf2xml` output. Content area are implicitly inferred by the cognitive skills of the reader. Moreover content areas can be serialized in many ways in the PDF. For instance, header and footer areas can be serialized before the document body area. The boundary delimitation of content areas inside a page is one of the main challenges.

Bounding the ToC areas over pages is not straight due to the absence of marks that separate them from other adjacent areas. In our process, positional information of headers and footers are inferred from the document structure in order to help the boundary delimitation of ToC areas. Taking into account the consistency of the styles within the ToC, and the style contrast with other parts should also help the delimitation.

We point out that there is no concept of “word” or “number” or “token” in PDF. In order to ease the processing, `pdf2xml` introduces the concept of “token”, a computational unit based on character spacing. In practice, output tokens correspond to words or numbers, what we can expect, but they can also correspond to a composition of several interpretable unit (e.g., “Introduction....5” or a breakdown of an interpretable unit (e.g., “C” “O” “N” “T” “E” “N” “T” ).

### 3.1.2 Locating the ToC pages

The ToC is located in the first pages of the document. It can spread over a limited number of contiguous pages. In the training set, we observed in practice up to three contiguous pages.

While observing various ToCs, it appears that few properties are common to all ToCs over the collection. Some ToCs have a title, others don’t have it. Some ToCs have section numbering, others don’t have it. One formal property is common to all ToCs we observed in the corpus: the page numbers of a ToC are *right-aligned* and form an increasing sequence of integers.

These characteristics are fully exploited in the core of our ToC identification process: we consider the pages of the first third of the document as a search space. Then we select the first right-aligned sequence of lines ending by an integer and that may spread over contiguous pages. We do not have to bound the expected number of ToC pages.

### 3.1.3 Building ToC entries

A ToC Entry is made of several parts, namely an optional level number, the title, an optional leader line (i.e., dotted line), and the page number. A regular expression is enough to capture the different part of the expected ToC entry. This process must be applied with care since there is a significant risk of confusion between two cases:

- long titles may spread over multiple lines, up to two lines in the corpus,
- major headings may not be associated to page numbers. Their page number is implicit and usually corresponds to the page number of the following subheading. For instance, when the title of a chapter is not specified in a ToC, its page number is the same as the page number of its first section.

Styling and span information helps managing these cases. Leader lines are optional and may not be present on all ToC entries, in particular on major headings. While leader lines ease the association between titles and page number when title is short or line spacing is thin, larger line spacing, eventually combined to larger font-sizes, can be enough to ease the association for the reader.

### 3.1.4 Inferring the Hierarchy

A ToC is a hierarchical structure. From a computational point of view, it can be seen as the result of a preorder depth-first tree traversal. In practice, it is not the case since we deal with natural language, not computational structure: all the titles do not have to be mentioned. It is the case for lower-level subheadings which could significantly burden the synthetic overview. It is also the case for the main title, or for unnamed parts, such as preliminaries, which are defined by their position and may be considered as minor parts.

A combination of contrastive effects usually reflects the hierarchy:

- larger *line-spacing* can be used to highlight major headings ;
- positive *indentation* can be used to indicate lower-level subheadings;
- *formatting character effects* such as bold, italic, character case and font-size can be used: smaller font-sizes or lower case for lower-level subheadings; bold or uppercase for higher-level headings;
- *numbering character sets*: uppercase letters (e.g., A, B, C, I, II) are more often used for numbering higher-level headings while lowercase letters (e.g., a, b, c, i, ii, iii, $\alpha$, $\beta$, $\gamma$) are used for lower-level subheading;
- *multi-level numbering structure*: subheading numbering (e.g., a, b, c) can be prefixed by parent numbering (e.g., A.2.a, A.2.b, A.2.c). The numbering of major parts, such as chapter (e.g., A), may not be prefixed in subheading multi-level number (e.g., 2.a, 2.b, 2.c) and may remain implicit.

Heading numbering may be prefixed by a functional term, such as Appendix, Chapter, Article, etc. It has to be handled. No specific list of terms has to be build. The term is repeated at the beginning of several ToC entries, before the heading number: it is enough to handle it.

In our process, the computation of the hierarchical structure is based on the combination of subheading indentation and multi-level numbering structure of ToC entries.

|  | Run | F-measure |
|---|---|---|
| Daniel | 1 | 42.72 |
| IHSMarkit | 1 | 39.41 |

Table 1: Results for the ToC Generation Task (test set)

| Xrx-measure Links |  |  |  | Title |  |
|---|---|---|---|---|---|
| Doc | Prec | Rec | F1 | Acc | book id |
| 0 | 97.7 | 48.6 | 64.9 | 84.5 | 1252823262 |
| 1 | 87.2 | 51.9 | 65.1 | 96.5 | 1139920265 |
| 2 | 22.2 | 40.0 | 28.6 | 91.9 | 0881817786 |
| 3 | 90.5 | 12.3 | 21.7 | 85.7 | 1150262910 |
| 4 | 100 | 10.4 | 18.9 | 42.4 | 0992626050 |
| 5 | 83.3 | 2.9 | 5.6 | 59.7 | 0949250459 |
| 6 | 100 | 12.4 | 22.1 | 94.6 | 1151059737 |

Table 2: Results for the ToC Generation Task on the test set

### 3.1.5 Computing the PDF Page Numbers

Once the ToC is built, each header is associated to a page number. This page number refers to the print version. The PDF page number we have to submit is slightly different: a page shift may appear if the first page of the PDF is not "page 1". It is the case when the document contains a title page, which might be unnumbered, or includes preliminary pages which might also be not numbered or might use a different numbering alphabet.

In order to get the appropriate PDF page numbers, we choose to compute the shift between PDF page numbers and printed page numbers. In order to extract printed page numbers, we select a sample of PDF pages. We then look for a series of integers located at the same position on different pages. Once we found this series, we get the page shift by calculating the difference between the first printed page number of the series and its corresponding PDF page number.

### 3.1.6 Results and discussion

The official results of our system Daniel on the test set are given in table 1. The detailed results of our system are given in table 2. As expected, the system always has a good precision and a lower recall. We point out that low precision for book 2 is due to the fact that the ToC of the prospectus is more detailed than the ToC of the groundtruth.

Good precision and low recall are linked to our method which is based on locating and parsing the ToCs. ToCs does not reflect the true structure of the prospectuses. They are generally less detailed: lower level headers are not included. Moreover, if no ToC is present or found, the system relies on a simple fallback.

Due to lack of time for implementation, we only handled ToC located on one-column page layout, which is the most common case for this kind of document. We did not handle the difference of page format for odd and even pages. Simple improvements can be done to cover these two cases.

As said at the beginning of this section the main improvements would come from taking into account the whole content of the document. We did not have enough time to handle it properly. It would allow the handling of documents without ToC and would permit the extraction of titles that are not included in the ToC. It would be particularly useful for these financial documents where fine-grain subdivisions are present but not represented in the ToC.

## 3.2 Title Detection

The very first feature one can think about is the length of the segment, titles are shorter segments and are seldom longer than a line. The second feature that came to our mind is that titles are likely to be nonverbal sentences and in general exhibit a simpler syntactical structure. Other features like those provided with the dataset can be useful: begins with numbering, material aspect (bold/italic), capitalization (begin with capitals, all_caps). We advocate that these differences are related to style, therefore the different baselines and systems we propose rely on stylistic features. We used the basic set of features given with the dataset and we added three other types of features:

**basic features** : Provided in the dataset (Begins with Numbering, Is Bold, Is Italic, Is All Caps, begin With Cap, Page Number)

**length** The length of the segment in characters

**stylo** Relative frequency of each punctuation sign, numbers and capitalized letters

Our other approach relies on character based features, used in particular in autorship attribution (Brixtel, 2015). We chose character n-grams because of their simplicity to compute. We try different possible values of $n$: $n_{min} \leq n \leq n_{max}$ with all possible $n_{min}$ and $n_{max}$ values between 1 and 10 (and $n_{min} \leq n_{max}$). We computed a relative frequency for each n-gram in each example to

|  | Cross-valid | Test-set |
|---|---|---|
| B1 (basic features) | 80.1 | 91.1 |
| B2 (basic + length) | 71.1 | 61.2 |
| B3 (stylo) | 75.5 | 87.6 |
| B4 (stylo+basic) | 72.2 | 84.2 |
| B5 (stylo+length) | 69.9 | 67.8 |
| B6 (stylo+basic+length) | 63.4 | 61.7 |
| n-grams ($1 \leq n \leq 1$) | 81.5 | 91.1 |
| n-grams ($1 \leq n \leq 2$) | 81.5 | 91.1 |
| n-grams ($1 \leq n \leq 3$) | 82.4 | 91.9 |
| n-grams ($1 \leq n \leq 4$) | 82.0 | 91.5 |
| n-grams ($1 \leq n \leq 5$) | 81.8 | 91.3 |

Table 3: Results for the title detection task for the Multinomial naive Bayes Classifier

|  | Cross-valid | Test-set |
|---|---|---|
| B1 (basic features) | 83.2 | 92.9 |
| B2 (basic + length) | 85.4 | 93.6 |
| B3 (stylo) | 85.4 | 93.2 |
| B4 (stylo+basic) | 90.4 | 94.2 |
| B5 (stylo+length) | 90.0 | 93.7 |
| **B6 (stylo+basic+length)** | 90.6 | **95.1** |
| n-grams ($1 \leq n \leq 1$) | 94.0 | 94.6 |
| n-grams ($1 \leq n \leq 2$) | 94.2 | 94.5 |
| n-grams ($1 \leq n \leq 3$) | 94.3 | 94.8 |
| **n-grams** ($1 \leq n \leq 4$) | 93.5 | **95.0** |
| n-grams ($1 \leq n \leq 5$) | 93.1 | 95.1 |

Table 4: Results for the title detection task for the DT10 Decision Tree Classifier (in bold our two submissions)

classify in order to take into account their various size. In fact, with absolute frequencies the results were significantly worse. We will only report results obtained with the Multinomial Naive Bayes (MNB) and the DT10 classifier since other classifiers did not offer better results than the DT10. SVM (with linear and non-linear kernels) had difficulties to converge with our baseline features due to their insufficent number.

In order to evaluate our methods and baselines we performed for each of them a ten-fold cross validation on the train set. The results on the train and test set are presented in Table 3 for the MNB classifier and Table 4 for the DT10 classifier. The first thing one can see is that the DT10 classifier outperforms the MNB in particular because the MNB classifier is not better with the stylometric features. The baselines with stylometric features worked well and our first submission was but the best method on the training data was the n-gram method (with $1 \leq n \leq 3$). However, we chose to submit the classifier trained with with $1 \leq n \leq 4$ because we believed it would be less prone to overfitting. With $n_{min} > 1$ or $n_{max} > 5$ the results drop significantly.

What we did not expect is that our best baseline performed much better on the test-set and was even better than our other submission. However, it is very interesting result since our experiments on the train set seemed to show that 1-grams were sufficient to build a reasonably efficient classifier.

### 3.3 Results and Discussion

We showed that very simple features can be of great interest, in particular in cases of training data scarcity. The methods we proposed can be improved in two different directions, regarding the features exploitation or exploring other features regarding the style of titles VS the style of non-titles. First, for improving a character-based approach it seems that LSTM architectures can be of great interest. The second option would be to extract syntactic patterns since sentence structures are quite different in titles.

## 4 Conclusion

Title detection and Table of Content (ToC) extraction are two important tasks for Document Analysis, in particular in the context of digital libraries and scanned books.

We proposed two types of features for the Title Detection task, we used a naive Bayses classifier as a baseline and a decision tree (DT10). We showed that simple stylometric features (frequency of punctuation, numbers and capitalized letters) combined with visual characteristics (bold, italic...) achieve better results than the best character n-gram approach (1-4 grams). Although this system did not achieved state-of-the-art performances, the results shows that simple and easy-to-compute features can provide very reliable results.

Regarding the ToC Extraction task, we choose to extract the structure from the ToC of the prospectuses. We are pleased to see that are our expectations are confirmed. Our system obtains a good precision and lower recall. For a next edition, we would like to focus on the extraction of the structure from the whole document content.

## References

Ali Ait Elhadj, Mohand Boughanem, Mohamed Mezghiche, and Fatiha Souam. 2012. Using structural similarity for clustering XML documents. *Knowledge and Information Systems*, 32(1):109–139.

Julian Alarte, Josep Silva, and Salvador Tamarit. 2019. What web template extractor should i use? a benchmarking and comparison for five template extractors. *ACM Trans. Web*, 13(2):9:1–9:19.

Joeran Beel, Stefan Langer, Marcel Genzmehr, and Christoph Müller. 2013. Docear's pdf inspector: Title extraction from pdf files. In *Proceedings of the 13th ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL '13, pages 443–444, New York, NY, USA. ACM.

Romain Brixtel. 2015. Maximal repeats enhance substring-based authorship attribution. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 63–71, Hissar, Bulgaria. INCOMA Ltd. Shoumen, BULGARIA.

Rik de Busser and Marie-Francine Moens. 2006. *Information extraction and information technology*, pages 1–22. Springer, Berlin, Heidelberg.

Sahar Changuel, Nicolas Labroche, and Bernadette Bouchon-Meunier. 2009. A general learning method for automatic titleextraction from html pages. In *Machine Learning and Data Mining in Pattern Recognition*, pages 704–718, Berlin, Heidelberg. Springer Berlin Heidelberg.

Béatrice Daille, Evelyne Jacquey, Gaël Lejeune, Luis Felipe Melo, and Yannick Toussaint. 2016. Ambiguity Diagnosis for Terms in Digital Humanities. In *Language Resources and Evaluation Conference*, Portorož, Slovenia.

Hervé Déjean. 2007. *pdf2xml open source software*. Last access on July 31, 2019.

Misha Denil, Alban Demiraj, and Nando de Freitas. 2014. Extraction of salient sentences from labelled documents. *CoRR*, abs/1412.6815.

Antoine Doucet, Gabriella Kazai, Sebastian Colutto, and Günter Mühlberger. 2013. Overview of the ICDAR 2013 Competition on Book Structure Extraction. In *Proc. of the 12th International Conference on Document Analysis and Recognition (ICDAR'2013)*, pages 1438–1443, Washington DC, USA.

Antoine Doucet and Miro Lehtonen. 2007. Unsupervised classification of text-centric xml document collections. In *Comparative Evaluation of XML Information Retrieval Systems, 5th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX*, volume 4518 of *Lecture Notes in Computer Science*, pages 497–509. Springer.

Emmanuel Giguet, Alexandre Baudrillart, and Nadine Lucas. 2009. Resurgence for the book structure extraction competition. In *INEX 2009 Workshop Pre-Proceedings*, pages 136–142.

Emmanuel Giguet and Nadine Lucas. 2010a. The book structure extraction competition with the resurgence software at caen university. In *Focused Retrieval and Evaluation*, pages 170–178, Berlin, Heidelberg. Springer Berlin Heidelberg.

Emmanuel Giguet and Nadine Lucas. 2010b. The book structure extraction competition with the resurgence software for part and chapter detection at caen university. In *Comparative Evaluation of Focused Retrieval - 9th International Workshop INEX, Vugh, The Netherlands, Revised Selected Papers*, volume 6932 of *Lecture Notes in Computer Science*, pages 128–139. Springer.

Silja Huttunen, Arto Vihavainen, Peter von Etter, and Roman Yangarber. 2011. Relevance prediction in information extraction using discourse and lexical features. In *Proceedings of the 18th Nordic Conference of Computational Linguistics (NODALIDA 2011)*, pages 114–121.

Stefan Klampfl, Michael Granitzer, Kris Jack, and Roman Kern. 2014. Unsupervised document structure analysis of digital scientific articles. *Int. J. Digit. Libr.*, 14(3-4):83–99.

Gaël Lejeune, Romain Brixtel, Antoine Doucet, and Nadine Lucas. 2015. Multilingual event extraction for epidemic detection. *Artificial Intelligence in Medicine*, 65(2):131 – 143. Intelligent healthcare informatics in big data era.

Gaël Lejeune and Lichao Zhu. 2018. A new proposal for evaluating web page cleaning tools. *Computación y Sistemas*, 22(4).

Thi-Tuyet-Hai Nguyen, Antoine Doucet, and Mickael Coustaty. 2017. Enhancing table of contents extraction by system aggregation. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 01, pages 242–247.

Sira Ferradans Rémi Juge, Najah-Imane Bentabet. 2019. The fintoc-2019 shared task: Financial document structure extraction. In *The Second Workshop on Financial Narrative Processing of NoDalida 2019*.

Dominika Tkaczyk, Andrew Collins, and Joeran Beel. 2018. Who did what?: Identifying author contributions in biomedical publications using naïve bayes. In *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries*, JCDL '18, pages 387–388, New York, NY, USA. ACM.

Yewei Xue, Yunhua Hu, Guomao Xin, Ruihua Song, Shuming Shi, Yunbo Cao, Chin-Yew Lin, and Hang Li. 2007. Web page title extraction and its application. *Information Processing & Management*, 43(5):1332 – 1347. Patent Processing.