UWB@FinTOC-2020 Shared Task: Financial Document Title Detection

Tomáš Hercig
NTIS – New Technologies
for the Information Society,
Faculty of Applied Sciences,
University of West Bohemia,
Technická 8, 306 14 Plzeň
Czech Republic
tigi@kiv.zcu.cz

Pavel Král
Department of Computer
Science and Engineering,
Faculty of Applied Sciences
University of West Bohemia,
Univerzitní 8, 306 14 Plzeň
Czech Republic
pkral@kiv.zcu.cz

Abstract

This paper describes our system created for the Financial Document Structure Extraction Shared Task (FinTOC-2020): Title Detection. We rely on the Apache PDFBox library to extract text and all additional information e.g. font type and font size from the financial prospectuses. Our constrained system uses only the provided training data without any additional external resources. Our system is based on the Maximum Entropy classifier and various features including font type and font size. Our system achieves F1 score 81% and #1 place in the French track and F1 score 77% and #2 place among 5 participating teams in the English track.

1 Introduction

Financial documents are used to report activities, financial situation, investment plans, and operational information to shareholders, investors, and financial markets. These reports are usually created on an annual basis in machine-readable formats often only with minimal structure information. The majority of these prospectuses are published without a table of content (TOC), which is usually needed to help readers navigate within the document.

The goal of the First Financial Document Structure Extraction Shared Task (FinTOC-2019) (Juge et al., 2019) was to analyse the financial prospectuses and automatically extract their structure similarly to Doucet et al. (2013). The Second Financial Document Structure Extraction Shared Task (FinTOC-2020) (Bentabet et al., 2020) adds French documents and greatly simplifies the data formats at the cost of not providing any text representation of the PDF files.

2 Task

The goal of FinTOC-2020 shared task is to extract the table of content from the financial prospectuses. Systems participating in this shared task were given a sample collection of financial prospectuses with different levels of structure and different lengths as training data. Data statistics for the title detection subtask are shown in Table 1.

The shared task can be divided into two steps: **1) Title detection** classifies given text blocks as titles or non-titles. **2) TOC generation** organizes provided headers into a hierarchical table of content.

We participated only in the Title detection subtask for both languages. For additional information (e.g. about TOC generation subtask) see the task description paper (Bentabet et al., 2020).

Label	French	English
Non-title	65.8k (90.8%)	186.3k (94.9%)
Title	6.6k (9.2%)	10.1k (5.1%)
PDF	47	52

Table 1: Data statistics for Title detection.

Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation, pages 158–162
Barcelona, Spain (Online), December 12, 2020.

We approached the title detection subtask as a binary classification task. For all experiments, we use Maximum Entropy classifier with default settings from Brainy machine learning library (Konkol, 2014).

3 Dataset

The provided training collection of documents contains the original documents in **PDF** format and annotations **JSON** file with gold labels. The JSON file consists of an array of TOC items representing each title with the following properties: **text** - text of the title, **id** - order of occurrence the title, **depth** - depth level of the title, and **page** - page of title occurrence.

In Table 1 we can see that title distribution among French and English data differs greatly, however, we don't know if the reason for this is that the documents have a different structure or a different approach to annotation was used.

4 Extraction

We decided to use the Apache PDFBox `https://pdfbox.apache.org/` version 2.0.20 to extract text and other metadata from the PDF files and then we use our own algorithm to link the annotations to the extracted text representation.

We consider each line of text a separate text segment and classify each segment as title or non-title. If there is a change in the font size or type we split the text into two lines. Additional metadata are extracted from the first occurring word of the given line. The metadata include the following features: Is_bold, Is_italic, Is_all_caps, Begins_with_cap, Begins_with_numbering, Left_position, Font_size, and Font_type. Note that some of these features were difficult to extract as there are more ways to create e.g. bold text in PDF format and the library does not provide a convenient interface to access e.g. vector elements.

5 Issues

The first shared task (FinTOC-2019) had some issues with the mapping of the XML text representation to the annotated CSV gold labels representation as reported by (Hercig and Král, 2019).

The second shared task (FinTOC-2020) removed the XML text representation and simplified the gold labels representation to JSON format, however, some of the problems still remained.

We did not get any annotation guidelines or explanation of some labels. It seems that the annotation process was incoherent - leaving us with different levels of depth and various parts of the title included or left out depending probably on the annotator of the current file or title.

We wrote an algorithm that tries to find the best mapping on a given page assuming the annotated text from the JSON training file appears in the same order of occurrence as the text extracted from the PDF file. Unfortunately, that is not always true, thus we decided to modify the training JSON files and fix the issues, described in the following sections, which caused our algorithm to fail. We manually fixed only the necessary part of the dataset in order for our algorithm to work.

5.1 Wrong Parameter

When we found a typo in the **id** or the **page** parameters in the JSON file we corrected the value according to the original PDF.

5.2 Missing Text Beginning

In some cases, the beginning of annotated text from the JSON file was missing. We fixed the occurrences our algorithm discovered. See the example below.

```
original:SUBSCRIPTIONS ...
fixed:(5) SUBSCRIPTIONS ...
```

5.3 Wrong Text Transcription

The JSON file contained wrong text transcription (e.g. additional spaces) that caused our mapping algorithm to fail on the given page because no match for the text was found. We corrected the text according to the original PDF.

6 Features

We tried to create the best feature set using all the extracted meta-information. The following features proved useful and were used in our submissions.

- **Character n-grams (ChN$_n$):** Separate feature for each n-gram representing the n-gram presence in the text. We do it separately for different orders $n \in \{1, 2\}$ and remove n-gram with frequency $f \leq 2$.
- **Binary Features (B):** We use separate binary feature for the following text characteristics (Is_bold, Is_italic, Is_all_caps, Begins_with_cap, Begins_with_numbering, Is_next_line_empty, Is_prev_line_empty).
- **Position Features (P):** We use four separate binary features to represent the difference in the left position of the text for two sentences. The positions can be equal, lower, greater, and missing. We compare sentence at position p with sentence at position $p-2$, $p-1$, and $p+1$.
- **First Orto-characters (FO):** Bag of first three orthographic[1] characters with at least 2 occurrences.
- **Last Orto-characters (LO):** Bag of last three orthographic[1] characters with at least 2 occurrences.
- **Font Size (FS):** We map the font size of text into a one-hot vector with length twelve and use this vector as features for the classifier. The frequency belongs to one of twelve equal-frequency bins[2]. Each bin corresponds to a position in the vector. We remove font sizes with frequency ≤ 2.
- **Font Size Diff (FSD):** We use four separate binary features to represent the difference in font size (FS) of the text for two sentences. The positions can be equal, lower, greater, and missing. We compare sentence at position p with sentence at position $p-1$ and $p+1$.
- **Font Type Diff (FTD):** We use three separate binary features to represent the difference in font type for two sentences. The font type can be equal, different, and missing. We compare sentence at position p with sentence at position $p-1$ and $p+1$.
- **Font Type Unigrams (FTU):** We tokenize font type name and use the presence of unigrams as a feature we remove unigrams with frequency $f \leq 1000$.

7 Results

The results in Table 4 show our ranking in the FinTOC-2020 shared task using the original test dataset. Our submissions and the fixed train datasets for both English and French are available for research purposes at `https://gitlab.com/tigi.cz/fintoc-2020`.

We performed ablation experiments to illustrate which features are the most beneficial (see Table 3). Numbers represent the performance change when the given feature is removed (i.e. lower number means better feature). We used approximately 30% of the fixed training dataset[3] for evaluation *(test)* and we used the rest of the dataset for training the features (see Table 2). We also repeated the experiment using *leave-one-out* cross-validation as the previous experiment seemed inaccurate. Our evaluation measure is macro-averaged F1-score.

We can see that the experiments are inconclusive as some of the findings are in contradiction. The most helpful features in terms of *leave-one-out* cross-validation apart from character bi-grams include both first and last orto-characters and font type unigrams. Last orto-characters, *FSD* and *FTD* were always beneficial. On the contrary position features and font size features were the least helpful features.

We believe that the reason that binary features were not very successful (except in French *test* setting) is that the extraction of these features was not accurate as mentioned in Section 4.

[1] All lower cased letters were replaced by "a", upper cased letters by "A" and digits by "1" (e.g. `"Char3" = "Aaaa1"`).

[2] The frequencies from the training data are split into twelve equal-size bins according to corresponding quantile.

[3] We used all annotations for ten and twelve JSON files for French and English respectively.

Language	French		English	
Label	Test*	Train*	Test*	Train*
Non-title	21.0k (31.9%)	44.8k (68.1%)	57.2k (30.7%)	129.0k (69.3%)
Title	1.7k (25.5%)	5.0k (74.5%)	3.4k (33.6%)	6.7k (66.4%)
Sum	22.7k (31.3%)	49.8k (68.7%)	60.6k (30.9%)	135.7k (69.1%)
PDF	10 (21.3%)	37 (78.7%)	12 (23.1%)	40 (76.9%)

Table 2: Dataset split for experiments.

Detailed statistical analysis into the datasets and gold labels for the test set would be needed in order to infer further, more accurate, insides.

Feature	F1-*test*	F1-*leave-one-out*
ALL*	89.15%	96.19%
ChN_1	-1.11%	-0.08%
ChN_2	0.54%	-3.54%
B	-1.67%	0.00%
P	-0.19%	0.02%
FO	-1.18%	-1.18%
LO	-1.50%	-0.90%
FS	0.14%	-0.24%
FSD	-0.61%	-0.11%
FTD	-0.64%	-0.24%
FTU	1.03%	-0.47%

* Using all features in the ablation study.

(a) French

Feature	F1-*test*	F1-*leave-one-out*
ALL*	87.74%	95.88%
ChN_1	0.70%	-0.09%
ChN_2	0.69%	-2.59%
B	1.51%	0.00%
P	0.38%	0.06%
FO	0.84%	-0.67%
LO	-0.78%	-0.97%
FS	1.68%	0.00%
FSD	-1.11%	-0.19%
FTD	-0.60%	-0.03%
FTU	-3.29%	-0.60%

* Using all features in the ablation study.

(b) English

Table 3: Feature ablation study.

Team	Submission	F1
UWB	1	81%
taxy.io	1	69%
Daniel	1	66%
DNLP	1	64%
Daniel	2	64%
Daniel	3	64%

(a) French

Team	Submission	F1
Amex	1	79%
Amex	2	79%
UWB	1	77%
Daniel	1	69%
Daniel	3	63%
Daniel	2	62%
DNLP	1	59%
taxy.io	1	55%

(b) English

Table 4: Results for Title detection.

8 Conclusion

In this paper, we described our UWB system participating in the FinTOC 2020 shared task.

Our best results have been achieved by the Maximum Entropy classifier combining available metadata, such as font type and font size, by careful feature engineering. Our system is ranked #1 in the French track and #2 among 5 participating teams in the English track.

Acknowledgements

This work has been partly supported by Cross-border Cooperation Program Czech Republic - Free State of Bavaria ETS Objective 2014-2020 (project no. 211).

References

Najah-Imane Bentabet, Rémi Juge, Ismail El Maarouf, Virginie Mouilleron, Dialekti Valsamou-Stanislawski, and Mahmoud El-Haj. 2020. The Financial Document Structure Extraction Shared task (FinToc 2020). In *The 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation (FNP-FNS 2020*, Barcelona, Spain.

A. Doucet, G. Kazai, S. Colutto, and G. Mhlberger. 2013. ICDAR 2013 Competition on Book Structure Extraction. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1438–1443, Aug.

Tomáš Hercig and Pavel Král. 2019. UWB@FinTOC-2019 shared task: Financial document title detection. In *Proceedings of the Second Financial Narrative Processing Workshop (FNP 2019)*, pages 74–78, Turku, Finland, September. Linköping University Electronic Press.

Rémi Juge, Najah-Imane Bentabet, and Sira Ferradans. 2019. The FinTOC-2019 Shared Task: Financial Document Structure Extraction. In *The Second Workshop on Financial Narrative Processing of NoDalida 2019*.

Michal Konkol. 2014. Brainy: A Machine Learning Library. In Leszek Rutkowski, Marcin Korytkowski, Rafal Scherer, Ryszard Tadeusiewicz, Lotfi Zadeh, and Jacek Zurada, editors, *Artificial Intelligence and Soft Computing*, volume 8468 of *Lecture Notes in Computer Science*, pages 490–499. Springer International Publishing.