UWB@FinTOC-2019 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-2019) Task A: Title Detection. We rely on the XML representation of the financial prospectuses for additional layout information about the text (font type, font size, etc.). 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 97.2% and is ranked #3 among 10 submitted systems.

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 goal of the Financial Document Structure Extraction Shared Task (FinTOC-2019) (Juge et al., 2019) is to analyse these financial prospectuses[1] and automatically extract their structure similarly to Doucet et al. (2013).

The majority of prospectuses are published without a table of content (TOC), which is usually needed to help readers navigate within the document.

2 Task

The goal of FinTOC-2019 shared task is to extract the table of content from the financial prospectuses. The shared task consists of two subtasks:

- Subtask A classifies given text blocks as titles or non-titles.
- Subtask B organizes provided headers into a hierarchical table of content.

We participated only in subtask A. For additional information (e.g. about subtask B) see the task description paper (Juge et al., 2019).

Systems participating in this shared task were given a sample collection of financial prospectuses with different level of structure and different lengths as training data.

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).

Data statistics for the title detection subtask are shown in Table 1.

Label	Test	Train
Non-title	13 928 (94.0%)	65 354 (86.4%)
Title	888 (6.0%)	10 271 (13.6%)

Table 1: Data statistics for Subtask A.

3 Dataset

The provided training collection of documents contains:

- PDF format of the documents
- XML representation of the PDFs as given by the Poppler utility libraries; this representation contains the text of the documents as well as layout information about the text (font, bold, italic, and coordinates).
- CSV file with gold labels.

[1] Official PDF documents in which investment funds precisely describe their characteristics and investment modalities.

Label	Test	Fixed Test	Train	Fixed Train
Non-title	13 928	12 844 (92.2%)*	65 354	60 533 (92.6%)
Title	888	821 (92.5%)*	10 271	10 209 (99.4%)
Sum	14 816	13 665 (92.2%)	75 625	70 742 (93.5%)

Table 2: Comparison of datasets with fixed issues.

The XML file consists of page elements and has essentially the following structure:

```
<page number="1" ...>
<fontspec id="0" size="11"
family="Times" color="#000000"/>
<fontspec id="1" size="9".../>
<text ...><b> </b></text>
<text ...>Man Umbrella SICAV </text>
...
</page>
...
```

The CSV file contains the following fields delimited by tabs. For more details see the task description paper (Juge et al., 2019).

- Text blocks: a list of strings computed by a heuristic algorithm; the algorithm segments the documents into homogeneous text regions according to given rules
- Begins_with_numbering: 1 if the text block begins with a numbering such as 1., A/, b), III., etc.; 0 otherwise
- Is_bold: 1 if the title appear in bold in the PDF document; 0 otherwise
- Is_italic: 1 if the title is in italic in the PDF document; 0 otherwise
- Is_all_caps: 1 if the title is all composed of capital letters; 0 otherwise
- Begins_with_cap: 1 if the title begins with a capital letter; 0 otherwise
- Xmlfile: the XML file from which the above features have been derived
- Page_nb: the page number in the PDF where the text block appears
- Label: 1 if text line is a title, 0 otherwise

According to the organizers, participants can either use the segmentation into text blocks suggested in the CSV file provided for the subtask A, or come up with their own segmentation algorithm which is highly encouraged.

We decided to use the XML file and thus needed to link the annotation labels to the original XML text representation.

4 Issues

We mentioned in previous section that the segmentation into text blocks is provided in the CSV file. However, that means that we need to find the mapping from the annotated text segments onto the original XML text representation.

We wrote an algorithm that goes through both files and tries to find the best mapping on a given page assuming the annotated text from the CSV file appears in the same order of occurrence as the text in the XML file. Unfortunately, that is not always true, thus we decided to modify the training CSV file and fix the issues, described in the following sections, that caused our algorithm to fail. We fixed only the necessary part of the dataset in order for our algorithm to work. The scale of these issues is illustrated in Table 2.

The percentage ratio in Table 2 is between the original and the fixed dataset. The star sign indicates that the labels were not known at the time and thus the issue described in Section 4.1 only eliminated duplicates not taking into consideration the assigned label, leading to the removal of more title labels compared to the train dataset.

The algorithm mentioned at the beginning of this section maps up to N text blocks from the XML file to one annotation. This is basically the reverse process to the one constructing the text blocks for the CSV file. We use the first matching text segment from the XML file to assign the font and other meta-information to the annotations.

The following example is the XML file text blocks that can be mapped to the example in Section 4.2.

```
<text ...><b>4. Stock exchange listing
 </b></text>
<text ...>The Sub-Fund ... </text>
<text ...>Details regarding ...
 ... Multi-Strategy. </text>
<text ...><b>5.   Shares </b></text>
```

4.1 Duplicate Entries

When we found a duplicate entry in the CSV file we removed the duplicity leaving only one occurrence of the text according to the original PDF. If the duplicate entries varied in the gold label we usually left the label indicating title.

In the following example we added the line number from the original CSV file delimited by colon and shortened the XML file name.

```
20139: General Meeting 0 0 0 0 1
 LU..._ManConvertibles.xml 24 1
20140: General Meeting 0 0 0 0 1
 LU..._ManConvertibles.xml 24 0
```

4.2 Wrong Order of Occurrence

The CSV file contains repetitions[2] of data causing our mapping algorithm to fail on the given page because of the wrong order of text occurrence. We corrected the repetitions leaving only one occurrence of the text according to the original PDF. If the duplicate entries varied in the gold label we usually left the label indicating title.

In the following example we added the line number from the original CSV file delimited by colon and left out the text characteristics, XML file name (`LU..._ManConvertibles.xml`), the page number (120), and parts of the texts as they are unnecessary. The bold text denotes the fixed version of the annotations.

```
21782:3. Currency ... 1
21783:The reference currency ...
 cannot be excluded. ... 0
21784:4. Stock exchange listing ... 1
21785:The Sub-Fund may apply ...
 Multi-Strategy. ... 1
21786:5. Shares ... 0
21787:The Sub-Fund shall ...
 Sub-Fund. ... 0
21788:6. Share classes ... 1
21789:General ... 1
21790:3. Currency ... 0
21791:The reference currency ...
 cannot be excluded. ... 0
21792:4. Stock exchange listing ... 0
21793:The Sub-Fund may apply ...
 Multi-Strategy. ... 0
21794:5. Shares ... 0
21795:The Sub-Fund shall ...
 Sub-Fund. ... 0
21796:6. Share classes General ... 0
```

4.3 Missing Text Beginning

In rare cases the beginning of annotated text from the CSV file was missing. We fixed the cases our algorithm discovered. See the example that occurred on line 21782 for XML file (`LU...ControlPFCo.xml`) below.

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

[2]We did not found these repetitions in the original PDF files nor in the XML files.

5 Features

We tried to create the best feature set using all the provided 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 all five text characteristics from the CSV file (Begins_with_numbering, Is_bold, Is_italic, Is_all_caps, and Begins_with_cap).
- **First Orto-characters (FO):** Bag of first three orthographic[3] characters with at least 2 occurrences.
- **Last Orto-characters (LO):** Bag of last three orthographic[3] characters with at least 2 occurrences.
- **Font Size (FS):** We map the font size of text into a one-hot vector with length ten and use this vector as features for the classifier. The frequency belongs to one of ten equal-frequency bins[4]. Each bin corresponds to a position in the vector. We remove font sizes with frequency ≤ 2.
- **Font Type Size (FTS):** For each font type we map the text length into a one-hot vector with length five and use this vector as features for the classifier. The frequency belongs to one of five equal-frequency bins[5]. Each bin corresponds to a position in the vector.
- **Text Length (TL):** We map the text length into a one-hot vector with length ten and use this vector as features for the classifier. The frequency belongs to one of ten equal-frequency bins[4]. Each bin corresponds to a position in the vector. We remove text lengths with frequency ≤ 2.

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

[4]The frequencies from the training data are split into ten equal-size bins according to 10% quantiles.

[5]The frequencies from the training data are split into five equal-size bins according to 20% quantiles.

6 Results

The results in Table 4 show our ranking in the FinTOC-2019 shared task using the original dataset.

Our submission UWB 1 was achieved using probability threshold $t = 0.8$ for the classifiers' predictions. The submission UWB 2 was achieved using the default threshold of $t = 0.5$.

Both submissions were outputs of our model trained on the fixed dataset and contained the fixed and the original test set.

For the original test data we used the predictions of our model trained on the fixed test file. Then the removed lines / labels from the original dataset were automatically matched to the fixed dataset and if an exact match was found for a predicted title we marked the removed line in the original test set as a title.

Our submissions and the fixed train / test datasets are available for research purposes at https://gitlab.com/tigi.cz/fintoc-2019.

We performed ablation experiments to illustrate which features are the most beneficial using the default threshold $t = 0.5$ (see Table 3). Numbers represent the performance change when the given feature is removed (i.e. lower number means better feature). We used approximately 20% of the fixed training dataset[6] for evaluation and we used the rest of the dataset for training the features. Our evaluation includes accuracy and macro-averaged F1-score which is slightly different from the task evaluation metric: weighted F1-score (see the python evaluation script provided by organizers).

We can see that all features are beneficial for the results. The most helpful features apart from character n-grams include binary features representing provided text characteristics from the CSV file, first orto-characters, and font size.

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

7 Conclusion

In this paper we described our UWB system participating in FinTOC 2019 shared task for financial document title detection.

[6] We used all annotations for five XML files.

Feature	Accuracy	F1-macro
ALL*	96.42%	94.07%
ChN $_1$	-0.60%	-1.08%
ChN $_2$	-3.88%	-7.52%
B	-1.69%	-2.23%
FO	-0.50%	-0.68%
LO	-0.30%	-0.31%
FS	-0.45%	-0.50%
FTS	-0.10%	-0.07%
TL	-0.13%	-0.06%

* Using all features in the ablation study.

Table 3: Feature ablation study.

Team	Submission	F1-weighted
Aiai	2	98.19%
Aiai	1	97.66%
UWB	2	97.24%
YseopLab	2	97.16%
FinDSE	1	97.01%
FinDSE	2	96.84%
UWB	1	96.53%
Daniel	1	94.88%
Daniel	2	94.17%
YseopLab	1	93.19%

Table 4: Results for Subtask A.

Our best results have been achieved by Maximum Entropy classifier combining available metadata, such as font type and font size, by careful feature engineering. Our system is ranked #3 among 10 participating systems' submissions.

Acknowledgments

This publication was supported by the project LO1506 of the Czech Ministry of Education, Youth and Sports under the program NPU I.

References

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.

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.