Document Functional Type Classification

Kristýna Němcová

[1] Konica Minolta Laboratory Europe, Brno
[2] NLP Centre, Masaryk University, Brno
kristyna.nemcova@konicaminolta.cz

Abstract. This paper presents methods used to classify documents into functional types (e.g. invoices, orders, scientific papers). We analyzed the current solution and we reproduced it with improvement. The problem is divided into classifications based on text and layout, then the results are combined. The work is applicable in office environment e.g. for searching according to a functional type. When appropriately combined with systems designed for a specific functional type, our work can contribute to the system performance.

Keywords: classification; documents; functional type

1 Introduction

In the business world, document processing is crucial. Knowing the functional document type facilitates work with each document. We can for example use different models for named entity extraction for marketing brochures and invoices. We identified thirteen functional types:

- Brochure
- Contract
- Financial Report
- Invoice
- Meeting minute, memo
- NDA
- Order (Purchase order)
- Patent
- Project charter (plan, gantt)
- Project status report
- Questionnaire
- Scientific article
- Technical Specification

Currently, we use the HyDoc functional type classifier. However, it has some issues described in Section 2. Moreover, we wanted to discover whether there are better methods to solve the classification.

Aleš Horák, Pavel Rychlý, Adam Rambousek (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2018, pp. 95–100, 2018.

1.1 Paper Outline

In Section 2, we describe the HyDoc classifier. Section 3 focuses on methods we have used, particularly on the layout-based classifier described in Section 3.2. Section 4 discusses the evaluation criteria. Section 5 contains final remarks.

2 Related Work

2.1 HyDoc Description

HyDoc document classifier is a system which is capable to perform document classification using both text and visual features (page layout). The system relies on two separate classifiers for the text and visual part; in particular, the visual part uses a convolutional neural network to classify a random sample of document pages: single-page classifications are then combined to yield a global estimate for the document class using Bayesian inference. A final ensemble neural network employing two hidden layers combines results from the text and visual part, providing an estimate for the document class. A set of confidences over all the classes are returned. [3]

2.2 Problems with HyDoc

As mentioned above, visual part of HyDoc classification system depends on random page samples. Randomization, that causes program to be non-deterministic, is a major flaw of their solution. Sometimes outcomes are diametrically dissimilar and it is hard to evaluate actual results.

3 Methods

In this section, we describe the process, we built in order to obtain the same or better results as the current HyDoc classifier. Similarly to HyDoc, we set up two classifiers, one taking text features, the other taking visual features.

3.1 Text-based Classifier

The text-based classifier is similar to HyDoc as the text-part is not problematic. We separated text from documents and applied fastText pre-trained word vectors trained on Wikipedia [2]. Simple two layer neural network was trained.

3.2 Layout-based Classifier

The classification based on document layout was a more complicated problem as there is a lot of research done in single page document image classification but almost none in purely multiple pages. This fact caused selection of combined single page classifier of document. The concept of transfer learning was used and we retrained ResNet-152 [1] for single page classification. We classify single pages and then combine the results. In addition, some document types have typically multiple pages, others have typically one page.

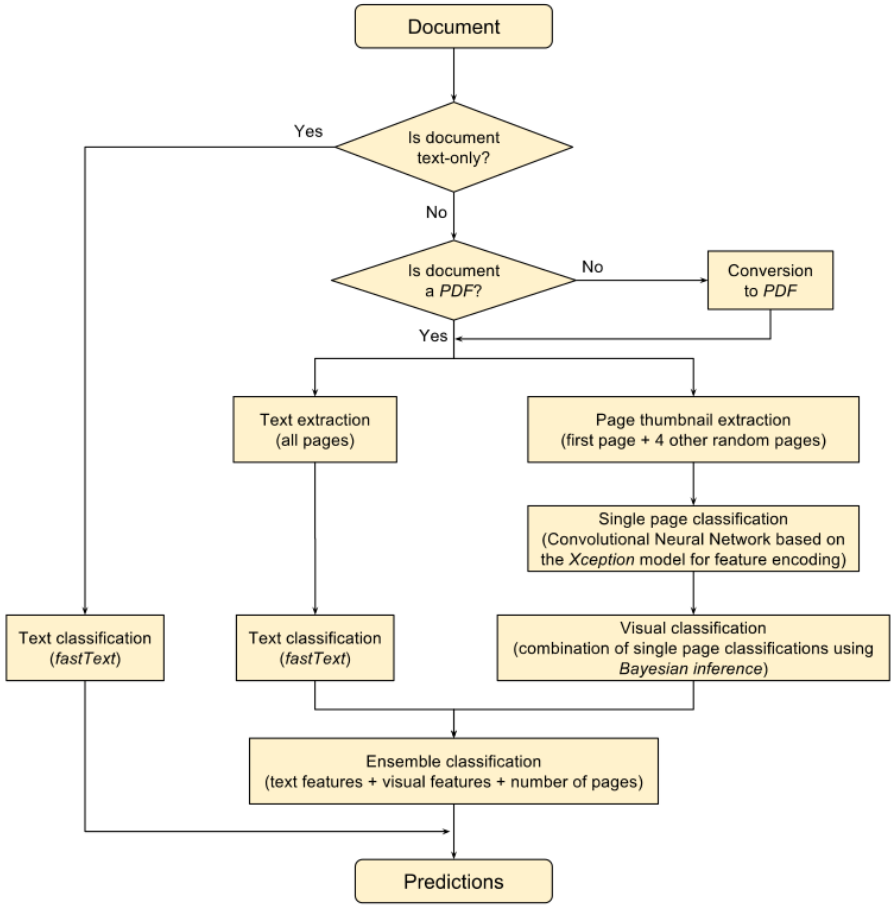

Fig. 1: Flow chart of HyDoc solution as provided by [3]

Possible Approaches for Layout-based Classification

Randomly Selected Pages HyDoc uses the questionable random selection with one classifier. The idea behind this could be the possibility of diverse document where pages are completely different. But that is unlikely to happen and therefore there are more cons than pros as described earlier.

Pages from Predefined Positions of the Document The correct approach seemed to be to take distinct pages in classification. The information about where the page is located in document and training only pages at the same position were vital. The unsophisticated solution would be to create multiple classifiers. The elegant answer was to convert images to different dimensions starting with three

(red, green, blue) and then add more dimensions. Nevertheless, this method has proven to be less promising presumably due to a small dataset.

All Pages of the Document The remaining solutions are based on one classifier. For instance, one can simply take all pages of the document. It works well on the train dataset. However, a problem occurred on validation and test dataset. There was recognizable over-fitting. After a closer look at the data, it is obvious why, because some of the documents have even more than thousand pages. Pages look very similar and therefore the classifier gives more weight to documents with more pages. Moreover it takes a lot of time to process such amount of images.

First Twenty Pages of the Document Experiments with number and position of pages were done (see Table 1) and the most promising approach was to take first twenty pages. It seems to be a compromise between a small dataset and massive over-fitting.

Fig. 2: Visualization of a data batch.

After a single page classification we need to combine the results on one document together as a outcome of layout-based classifier. So far, the used approach is to simply add probability matrices into a new matrix.

3.3 Final Classifier

The final classification is done by a meta-classifier. It takes results from text and layout-based classifiers as inputs of a simple neural network. Nonetheless, the problem with final classifier centers around a small dataset as we used 80 % of our data for training the previous classifiers.

4 Evaluation

The dataset contains at least hundred documents per each of the 13 functional types which means around 80 training documents per class. That is quite a small

amount. In a single image classification, we overcame this problem as we use up to 20 pages therefore the dataset become sufficient. Although as described above, this problem appears especially with combining results from text and layout.

The text data from documents provide reliable results as expected. We experimented with the neural network architecture and finally got the accuracy of 0.910.

As the decision to use only one classifier for all pages in layout-part was done, we needed to decide what pages to take into consideration. The best result came with first twenty pages combined with L2 penalty to prevent over-fitting [4]. The accuracy is 0.548.

Table 1: The results of approaches with one classifier

Pages	Accuracy
first page only	0.478
1st, 2nd, 2 pages from the middle, last page	0.521
all pages	0.532
first twenty pages	0.548

5 Conclusion and Future Work

In this paper, we have shown methods for classifying documents based on their functional type. We described an existing solution and examined its flaws. Our own approach based on two separate classifiers for text and layout was introduced. The text-part is robust and straightforward. In the layout-part, the single page classifier was properly examined and completed.

For the final layout-based classification, single page probability matrices are added. A more complex solution would be to compute confidence of matrix and add only few or let them vote or even to train a meta-classifier. Future experiments will show what is better.

Final prediction of functional types combines text and layout-based classifiers. The dataset for this part is small. The question is whether to include number of pages as a feature. It could provide a valuable insight, but also bias the outcome due to the data.

Given the current result, we can fearlessly say that our final classifier will have accuracy around 90% as the most weight is lying on the text-part. The layout-part will provide higher stability in cases of visually recognizable documents.

Acknowledgements. This research was supported by Konica Minolta Laboratory Europe.

References

1. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. ArXiv e-prints (Dec 2015)
2. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. CoRR **abs/1607.01759** (2016), http://arxiv.org/abs/1607.01759
3. Konica Minolta Laboratory Europe: HyDoc: Software Manual. Version 1.0. Unpublished
4. Krogh, A., Hertz, J.A.: A simple weight decay can improve generalization. In: Proceedings of the 4th International Conference on Neural Information Processing Systems. pp. 950–957. NIPS'91, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1991), http://dl.acm.org/citation.cfm?id=2986916.2987033