OpenPSS: An Open Page Stream Segmentation Benchmark

Ruben van Heusden[1(✉)], Jaap Kamps[2], and Maarten Marx[1]

[1] Information Retrieval Lab, University of Amsterdam, Amsterdam, The Netherlands
{r.j.vanheusden,maartenmarx}@uva.nl
[2] Faculty of Humanities, University of Amsterdam, Amsterdam, The Netherlands
kamps@uva.nl

Abstract. In recent years, an increasing number of companies and institutions have begun the process of digitizing their physical records to promote digital access and searchability of their collections. For cost-efficiency, documents are often scanned in consecutively, resulting in large PDF files consisting of many documents. Although cost-effective, this practice can be harmful for searchability when these concatenated documents are used to build a search engine. The task of Page Stream Segmentation is concerned with recovering the original document boundaries through the analysis of the text and/or images of these PDF files. Currently, many of the approaches to solving this problem make use of machine learning techniques that require significant amounts of training data. However, due to the sometimes sensitive nature of the data, few large datasets exist, and there is a lack of agreed-upon metrics to measure system performance.

In an effort to resolve these issues and provide a comprehensive overview of the state of the field, we constructed the *OpenPSS* benchmark, consisting of two large public datasets and a comprehensive study of various types of approaches, evaluated using multiple evaluation metrics. The datasets originated from several Dutch government institutions, cover a heterogeneous set of topics, and total roughly 141 thousand pages from around 32 thousand documents.

The experimental results show that ensemble methods using both the text and image representations of pages are superior to uni-modal methods, and that image-based neural methods are not as robust as text models when evaluated on out-of-distribution data.

Keywords: Page Stream Segmentation · benchmark · text classification

1 Introduction

Through the advent of modern technologies such as the internet, users can have access to vast amounts of information, including information contained in documents that were previously only accessible as physical records. This development

A. Antonacopoulos et al. (Eds.): TPDL 2024, LNCS 15177, pp. 413–429, 2024.
https://doi.org/10.1007/978-3-031-72437-4_24

has captured the interest of many companies and institutions, who are now starting to digitize their physical records to promote access to their collections. Examples of this include the digitization of library archives, and the online publication of legal records and court proceedings [4,8,26]. One of the steps in the digitization process concerns the scanning of documents into electronic formats such as PDF files, so that they can be published online. In practice, documents are often scanned in consecutively for convenience, resulting in large PDF files consisting of multiple documents. Although this might seem innocuous, this practice can have severe consequences on downstream tasks, such as when incorporating these documents in a search engine. As the atomic units in search engines are often documents, if an index of the collection is build using concatenated documents, it is possible that relevant documents are not scored properly as they might be contained within longer documents and overshadowed by irrelevant content.

Although work has recently been done on applying state-of-the-art machine learning techniques such as BERT [7] and LSTM [12] models to the task of PSS [4,8,26], the comparison of the efficacy of these models is complicated, as most models are evaluated on private datasets with differing evaluation setups. Moreover, the few datasets that are publicly available are either small in size or lack the presence of both the text and image of the pages, making comparisons across methods that use different modalities difficult. This problem is compounded by the fact that there is no clear, standard evaluation setup for the task, but that different evaluation metrics are used depending on the type of approach taken or the intended downstream task.

In an attempt to mitigate the aforementioned issues and provide a clear overview of the field, we present the *OpenPSS* benchmark, consisting of two large PSS datasets acquired from Dutch Freedom of Information Act (FOIA) requests. We use these two datasets to evaluate a wide range of approaches discussed in the literature using a uniform set of evaluation metrics, and explore various aspects of the PSS task, such as model ensembling and the robustness of various methods to out-of-distribution data.

The main contributions of this work can be summarized as follows: (1) We release the OpenPSS benchmark, consisting of two large annotated datasets with both the images and text of pages. (2) We provide an extensive overview of the performance of a multitude of segmentation models and methods, including different model ensemble strategies and the robustness of the models on out-of-distribution data. (3) We provide a brief analysis of the practical implications of applying a PSS method to split documents for use in a search engine.

The rest of the paper is organized as follows. Section 2 introduces the related work regarding several aspects of page stream segmentation and the current state-of-the-art. Section 3 discusses the construction of the dataset and the different tasks and approaches used in this paper. Section 4 presents the main results of the conducted experiments, followed by an analysis of the results when applied in the setting of a search engine in Sect. 5. We finish with a discussion and conclusion in Sects. 6 and 7.

2 Related Work

2.1 Page Stream Segmentation

The task of PSS can be seen as a particular instance of the broader fields of *Stream Segmentation* or *Information Segmentation*, which involve segmenting information from various modalities into either semantically, topically or syntactically coherent units. Within these broader fields, the field of *text segmentation* is most closely related, as it deals with the segmentation of textual units of various granularities, and methods developed for text segmentation problems can often be readily transferred to be applied in PSS. As such, methods such as Support Vector Machines (SVMs) and Multilayer Perceptrons combined with word embeddings have been used in early approaches to the task [2,6,9,19]. Most of these papers approach the task as a binary classification task, where the model is tasked with, for each page, determining whether or not it starts a new document, with pages represented as a set of word embeddings, TD-IDF vectors, or some other textual representation.

With the introduction of modern neural machine learning algorithms such as BERT and VGG16, the performance on the PSS task has increased significantly, although the general approach of binary classification on pages has remained the same. In contrast to text segmentation, PSS methods usually have the images of the pages available, and therefore models from computer vision (such as VGG16) can also be applied to the task. Wiedemann and Heyer [26] introduce a neural segmentation model based on convolution neural networks to segment page streams, classifying each individual page. Separate models are created for both the text and image of the pages (using word embeddings for the text domain input) and their performance is compared. The methods are compared using the Tobacco800 [1,17] dataset and the private *Archive26K* dataset. The models are evaluated using page-level precision, recall and F1 scores, and all of them outperform baselines based on SVMs. Experiments with combining both the image and text models showed that the combination of the modalities yielded the best performing model on both datasets. Later work by Braz et al. [4] also adopted this binary classification approach, but instead focused only on the image domain. Using the EfficientNet [24] architecture instead of an VGG16 model they improve upon the scores of Wiedemann and Heyer on the Tobacco800 dataset. In a similar vein, Guha et al. [8] replaced the text model from Wiedemann and Heyer with a BERT model, and report improvements for both the uni-modal setting as well as the performance of an ensemble containing the BERT model and a VGG16 model on the page-level precision, recall and F1 scores.

Although recent developments in PSS have mostly focused on binary classification of pages, several text segmentation methods instead treat the segmentation task as a sequence labeling task, where a complete sequence is inputted, and the model outputs predictions for each input simultaneously. One of the first to use this approach were Hernault et al. [10], who used conditional random fields for discourse segmentation and outperformed several methods that were state-of-the-art at the time, including SVMs. Later papers have tried different

methods [14,18,25], with for example Koshorek et al. [14] introducing a model based on the LSTM architecture and evaluating their method on *WIKI-727K*, a dataset consisting of Wikipedia articles, where the task is to separate the different sections of a Wikipedia article. Although the granularity of segmentation is different for PSS, i.e. pages instead of sentences or paragraphs, the basic principle remains the same, and thus this technique can also be applied to the task.

2.2 Other Datasets

Although most PSS datasets are private, two public datasets are available, namely the *Tobacco800* [27,28] and *AI Lab Splitter* [4] datasets. The Tobacco800 dataset consists of a single stream of 800 documents, totalling almost 1300 black-and-white pages, and consists of documents released through court proceedings against several large tobacco firms. The language used in most of these documents is English, and the dataset contains both the images in 300 DPI PNG format, as well as text extracted using Optical Character Recognition (OCR) techniques. The A.I. Lab splitter dataset consists of 1,869 streams, and has approximately 32,000 pages, originating from court proceedings from Brazilian courts, mostly in Portuguese, containing only the images of the pages in 224 by 224 pixel format, with text not being included. Due to the low resolution at which the images were saved, the text could not be extracted using OCR techniques.

The OpenPSS benchmark presented in this paper is an extension of previous work [11] where a single smaller dataset was used, and some preliminary experiments using only non-learned baselines were performed comparing different metrics. This paper expands on that work by including a large variety of segmentation methods, expanding the size of the original dataset and adding the OpenPSS-LONG dataset to enable out-of-distribution experiments.

3 Method

3.1 Datasets

The OpenPSS benchmark consists of two large datasets, both of which consist of documents released on the request of citizens as part of the Dutch Freedom of Information Act (FOIA). This act requires various levels of the government to release information regarding their decision-making process to the public. As these requests can be very broad and can cover a wide range of topics, the documents in the datasets are very heterogeneous, ranging from official letters and meeting reports to email threads and screenshots of WhatsApp messages. Figure 1 contains the pages of two documents from a larger page stream, where the pages also contain some redactions, as parts of the documents are confidential and not all information can be released. The two datasets in the benchmark are comparable in terms of the number of one-page documents (30% and 31%) but quite different in terms of the length of the streams, which is why we refer to them as the OpenPSS-LONG and OpenPSS-SHORT datasets.

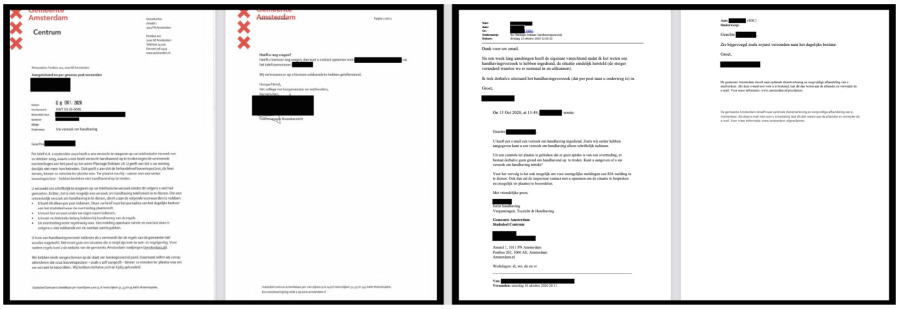

Fig. 1. Example of a page stream from the OpenPSS-SHORT dataset with two documents, both consisting of two pages, with the black borders indicating document boundaries

The OpenPSS-LONG dataset originated from requests to ministries during the COVID-19 pandemic, and were manually annotated with document boundaries. The OpenPSS-SHORT dataset is constructed from three Dutch governmental bodies who published the released archives as zip files, and thus the true documents were known. To turn these zip archives into streams, the original documents were concatenated into one large PDF file in order of appearance in the zip file and the boundary pages were recorded, similar to the approach taken by Reynar [21] and Choi [5].

Approximately one-third of all pages were scans without underlying text, and the underlying text of the other pages was often of poor quality. To address this issue, OCR was performed on both datasets to extract the text, using Tesseract (version 5)[1] with the Sauvola binarization algorithm [22].

Table 1 shows the main statistics of both the OpenPSS-LONG and OpenPSS-SHORT datasets, as well as those of the *Tobacco800* and *A.I. Lab Splitter* datasets, the only other two publicly available PSS datasets.

Table 1. Overview of the key properties of the OpenPSS-LONG, OpenPSS-SHORT, Tobacco800 and A.I. LAB Splitter datasets

	Number of Streams	Number of Documents	Number of Pages	Percentage of 1 Page Documents	Median Number of Pages in Stream	Median Number of Documents in Stream	Image+Text
Tobacco800	1	736	1,290	0.63	-	-	✓
AI LAB Splitter	1,869	5,487	31,789	0.46	9	1	×
OpenPSS-LONG	110	24,181	89,491	0.30	217	55	✓
OpenPSS-SHORT	312	8,162	52,177	0.31	60	8	✓

In the two existing datasets, and in particular in Tobacco800, single-page documents are over-represented. Because of this, so-called 'degenerate' segmentation algorithms [3], or algorithms that simply predict no boundaries or only boundaries can achieve deceptively high scores. Since Tobacco800 only consists

[1] https://github.com/tesseract-ocr/tesseract.

of a single stream, partitioning the stream such that each page is a document results in a (boundary page) recall of one, a precision of .63, and thus a rather high F1 score of .77. Even though the A.I. Lab splitter dataset contains a large number of streams, nearly half of them still consist of only one document. In this case, the opposite degenerate algorithm consisting of"do not split at all" yields a precision of 1, a recall of almost .5 and thus an F1 score of roughly .60.

3.2 PSS Variants

In this paper we distinguish between two types of segmentation tasks, namely *Standard PSS* and *Robust PSS*. Standard PSS is the classic segmentation task where, given an input stream S of pages, the task is to partition S into consecutive non-overlapping page-sequences (the documents). Robust PSS is similar to standard PSS, except that the algorithms are tested on out-of-distribution data, in this case from a different provider than the dataset used for training. The Robust PSS task better resembles PSS in practice, as a system trained on a specific dataset might well be used on other datasets, such as a system developed for one library being employed for a different library or a general model integrated into a search engine.

3.3 Models

Baselines. As fixed non-learned approaches to PSS can score remarkably well, the so-called "degenerate algorithms" [3] are included in the experiments. These are: each page a document; the whole stream one document; fixed size segments based on the mean or median document length (measured either on the corpus or on the stream level). In the paper these are referred to as *Singleton Documents* , *Giant Document* and *Mean Document Length* respectively.

Strong Simple Baselines. To investigate the effectiveness of simple non-parametric and non-linear learning algorithms, the K-nearest neighbors (KNN) and XGBoost algorithms are included as baselines, with separate versions for the text and image domains, as well as a version that combines both modalities (referred to as KNN-Ensemble and XGBoost-Ensemble). For this multimodal version that representations of both modalities are simply concatenated and passed to the model. For both the KNN and XGBoost algorithms, implementations from *scikit-learn* with the default parameter settings were used.

Neural Methods. For the evaluation of neural methods on the benchmark, a selection of models from recent work has been taken, namely the TEXT-CNN and VGG16 methods from Wiedemann and Heyer [26], the BERT model from Guha et al. [8] and the EfficientNet from Braz et al. [4]. The TEXT-CNN model consists of a GRU model followed by a CNN, where word embeddings are fed into the GRU unit to create page representations, and the convolutional neural network makes a binary classification based on this vector. The GRU model has a hidden dimension of 128, both the GRU and CNN have a learning rate of

0.0001 and the model is trained for 20 epochs. The VGG16 model is pretrained on the ImageNet dataset, taking the raw pixels of an input image and outputting a binary classification. The model was trained with 100 convolutional filters of sizes 3, 4 and 5, a learning rate of 0.0001 and a batch size of 128. The BERT model from Guha et al. takes as input the raw text of a page, and classifies a page by inputting the CLS token to a final linear layer. Since pages can exceed the maximum token length of 512, documents that are longer are truncated by taking the first and last 75 tokens. The model is trained for 100 epochs with a learning rate of 0.00001 and a batch size of 8. To be usable for Dutch, a Dutch version of BERT was used in the experiments[2]. The approach from Braz et al. using the EfficientNet architecture is similar to the VGG16 method, where the VGG16 model is replaced with the EfficientNet architecture. The training parameters of the model are identical to that of the VGG16 model.

Sequence Labelling Methods. Inspired by the work of Koshorek et al. [14], an LSTM-based sequence classification algorithm is included in the experiments, to investigate whether such a method, where the predictions of different pages can influence each other, would perform well on the task. A similar method to the paper of Koshorek et al. is used, where either pretrained image vectors or Doc2Vec vectors are used as input to the model, and the model outputs a sequence of binary classifications for each page. The LSTM has a total of 128 hidden units, 1 layer and is trained for 100 epochs with a learning rate of 0.001. As the input to the model is a stream, which can potentially be very long, the model is fed segments of 64 pages at a time, to mitigate the known issues with LSTMs and long-term dependencies.

Representations. All binary classification algorithms (except for KNN , XGBoost and TEXT-CNN) make use of the raw input text, or the raw pixels of the input image. The TEXT-CNN model uses word embeddings trained for Dutch, and the XGBoost and KNN use either features extracted from a pretrained VGG16 model (for the image domain), or page embeddings extracted from a Dutch Doc2Vec model [16] (for the text domain). The image representations have 2,048 dimensions and the text representations have 300 dimensions.

The datasets and code required to reproduce the experiments in this paper are publicly available on Github[3].

3.4 Model Ensembling

As shown in previous work, methods that combine both the image and the text modalities usually yield improved performance over their uni-modal counterparts. To investigate the effect of combining models from different modalities, we compare two strategies for ensembling models, commonly referred to as *early ensembling* and *late ensembling*, which differ in the manner in which the two

[2] https://github.com/wietsedv/bertje.
[3] https://anonymous.4open.science/r/OpenPSSbenchmarkTPDL-D851/.

modalities are combined. In early ensembling, the final layers of two models are combined before the final classification is made, and a prediction is obtained by adding a final classification layer on top of this combined output to obtain the final prediction [20].

In late ensembling, the output probabilities of the models after the softmax operation are combined to output the final prediction. In this work, a simple linear combination of the output vectors of two models is used to obtain a single output from the model. To obtain an estimate on the theoretical performance of an ensemble, the maximum achievable scores of that model can be calculated by following work by Kuncheva [15], where the output of the ensemble model is considered correct as one of the models is correct, providing an upper bound on the performance of the ensemble.

3.5 Evaluation

Model performance is reported by using the standard confusion table based metrics, precision, recall and their harmonic mean F1. These metrics are reported both at the level of pages and at the level of documents. The page level metrics are commonly used in PSS and have a straightforward interpretation. However, PSS is not a page classification, but a document segmentation task, and thus a metric devoted to this task may provide a better estimate of the performance of a model, as page-level metrics for example to not distinguish between severity of errors.

For the document level evaluation, we report the Panoptic Quality (PQ) score, developed in computer vision [13]. PQ is an F1 score for partial document matching in which the score is weighted by the amount of overlap between a true and predicted document. The overlap between a true document t and a predicted document p, both of which are seen as sets of pages is measured by their Jaccard similarity and is called *Intersection over Union* $IoU(t, p)$. A pair (t, p) is a True Positive if $IoU(t, p) > 0.5$. Note that this constraint enforces at most one True Positive pair for each true or predicted document. Let TP be the set of True Positives. Then the set of False Positives FP consists of all predicted documents p which are not part of a True Positive pair and similarly, $t \in FN$ iff t is not part of a TP pair. Now the document level precision, recall and harmonic mean F1 can be defined as usual. Kirillov et al. also propose weighted versions of these scores which are obtained by multiplying them by the average IoU of the True Positives. This last measure is called the *Segmentation Quality (SQ)*.[4]

When reporting results we will report precision, recall and F1 measured at the page-level, and the weighted and unweighted F1 scores measured at the document level (referred to as Unweighted Document F1 and Weighted Document F1), together with the Segmentation Quality SQ. All metrics are always calculated per stream. As the test sets consist of multiple streams, we measure the performance of models by the averages of the metrics over the streams.

[4] Kirilov et al. call the unweighted F1 the recognition quality RQ, and the weighted $F1$, which equals $RQ \times SQ$ the Panoptic Quality PQ.

4 Results

4.1 Standard Page Stream Segmentation Task

The main results of the *Standard PSS* task are shown in Table 2, where the models are grouped into their approach types as described in Sect. 3.3, and evaluated on both the page- and document-level metrics. The neural PSS approaches outperform all the other models on document-level metrics on both datasets, with the BERT-EfficientNet ensemble achieving the best performance. As Table 2 contains a lot of information, Fig. 2 shows a summary of the main results on both datasets for the Weighted Document F1 score, sorted on their average performance on both.

Although the neural methods are the best performing class for both datasets, the KNN and XGBoost baselines produce competitive numbers given their simplicity, and for the page-level metrics they outperform the neural models on page-level precision and recall for a few variations. This result is relevant to real-world applications, as the simple baselines are cheap to compute, and can still prove competitive under these constraints.

For both the simple baselines and the neural methods, the combination of the modalities produces the best results, on both the OpenPSS-LONG and OpenPSS-SHORT datasets, where the BERT-EfficientNet late-ensembling approach that combines the output probabilities of both models outperforms the early-ensembling technique. The best-performing combination, BERT and EfficientNet late-ensembling, has not yet been tried in the literature, but is, on this benchmark, the state-of-the-art approach. Note that for the KNN and XGBoost methods, the ensembling method consists of simply concatenating and scaling input features. This simple strategy proves effective, as it outperforms the unimodal approaches for the the KNN and XGBoost methods.

.The TEXT-CNN and BERT models outperform the VGG16 and EfficientNet models on the OpenPSS-LONG dataset, but for the OpenPSS-SHORT dataset the image models outperform their textual counterparts. Similarly for the KNN and XGBoost models, the text-based models marginally outperform the image models on the OpenPSS-LONG but the image-based models perform slightly better on the OpenPSS-SHORT dataset.

The brief investigation into the sequence labelling approaches for PSS shows that this approach does currently not stand up to the state-of-the-art binary classification methods. Although the LSTM-VGG16 model produces results similar to that of the XGBoost and KNN methods on the OpenPSS-SHORT dataset, the results are not nearly as good for the model based on Doc2Vec embeddings, and both models perform poorly on the OpenPSS-LONG dataset.

The main reason for the subpar performance of the LSTM-based methods is the length of the stream that has to be classified, corroborated by the fact that both methods perform worse on the OpenPSS-LONG dataset. The low performance of both LSTM-based methods leads us to conclude that these models are currently not consistent enough for the task of PSS and therefore will not be included in further robustness experiments, as their baseline performance is simply too low.

Table 2. Results of the standard PSS task for the various algorithms on the OpenPSS-LONGand OpenPSS-SHORT. Scores reported on the page- and document level. All scores are calculated per stream; the reported scores are the averages over the scores of the streams

	OpenPSS-LONG						OpenPSS-SHORT					
	Page			Document			Page			Document		
Model	P	R	F1	SQ	F1	Weighted F1	P	R	F1	SQ	F1	Weighted F1
Non-Learned Baseline												
Mean Document Length	0.38	0.48	0.42	0.77	0.33	0.27	0.43	0.51	0.47	0.71	0.48	0.39
Singleton Documents	0.33	1.0	0.47	0.91	0.18	0.18	0.26	1.0	0.39	0.56	0.09	0.09
Giant Document	1.0	0.08	0.11	0.10	0.06	0.06	1.0	0.31	0.41	0.31	0.26	0.21
Strong Simple Baselines												
KNN-VGG16	0.69	0.73	0.66	0.85	0.54	0.50	0.84	0.63	0.66	0.81	0.60	0.55
KNN-BERT	0.63	0.77	0.66	0.86	0.55	0.51	0.73	0.60	0.60	0.81	0.55	0.50
KNN-Ensemble	0.65	0.77	0.68	0.84	0.58	0.54	0.83	0.65	0.68	0.81	0.61	0.57
XGBoost-VGG16	0.72	0.73	0.68	0.87	0.58	0.54	0.84	0.63	0.67	0.81	0.60	0.54
XGBoost-BERT	0.74	0.67	0.66	0.86	0.55	0.51	0.77	0.58	0.61	0.76	0.54	0.48
XGBoost-Ensemble	0.74	0.73	0.70	0.87	0.61	0.57	0.85	0.66	0.68	0.83	0.62	0.57
Visual Representations												
VGG16	**0.86**	0.70	0.72	0.92	0.64	0.61	0.81	**0.77**	0.74	0.88	0.68	0.64
EfficientNet	0.82	0.85	0.80	0.92	0.76	0.73	0.83	0.75	0.75	0.89	0.71	0.68
Textual Representations												
TEXT-CNN	0.81	**0.88**	0.81	0.91	0.78	0.75	0.81	0.76	0.73	0.86	0.67	0.63
BERT	0.84	**0.88**	**0.83**	0.91	0.79	**0.77**	0.81	0.73	0.72	0.86	0.66	0.62
BERT-EfficientNet Combination												
Early Ensembling	0.82	0.86	0.81	0.92	0.77	0.74	0.83	0.76	0.75	0.88	0.70	0.66
Late Ensembling	0.85	**0.88**	**0.83**	**0.93**	**0.80**	**0.77**	**0.87**	0.76	**0.76**	**0.90**	**0.72**	**0.69**
Sequence Labelling Methods												
LSTM-VGG16	0.51	0.50	0.48	0.78	0.41	0.37	0.75	0.61	0.63	0.76	0.57	0.53
LSTM-BERT	0.38	0.57	0.42	0.79	0.30	0.27	0.53	0.66	0.55	0.74	0.47	0.32

4.2 Robust Page Stream Segmentation Task

In the robust experiment, all the models are trained on one dataset, and tested on the other, and their performance is compared to their performance when trained and tested on the same dataset. Table 3 shows the main results of the robustness experiments, and Figure 3 shows a condensed overview, showing the relative performance drop of the methods when trained and tested on a different dataset.

For the neural methods, the text-based models are the most robust, with the TEXT-CNN model achieving the highest scores on both the OpenPSS-LONG and OpenPSS-SHORT datasets. The ensemble methods do not perform as well as the text-only methods, but they outperform the image-based models when averaged over the two datasets.

The difference between the robustness of the text and image models can be explained by the fact that the textual representation of pages varies less across different corpora, and that text-based models are able to use more general features to distinguish between pages, such as language usage, or implicit document types to distinguish between pages. However, this is less the case for the image

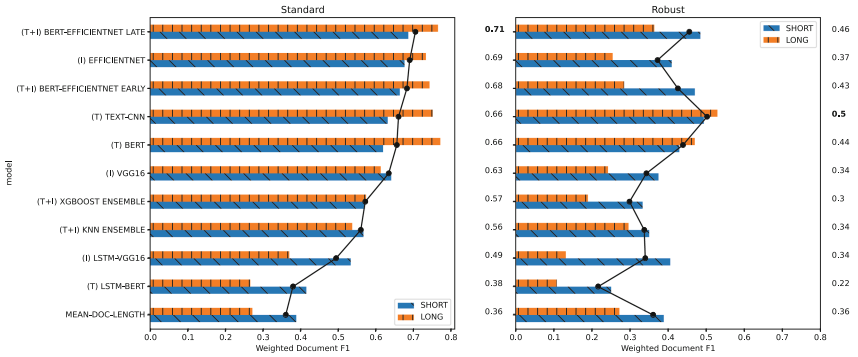

Fig. 2. Weighted Document F1 scores for selected models on the Standard PSS task (left) and the robust PSS task (right) for both the OpenPSS-SHORT and OpenPSS-LONG datasets. Bold indicates the model with the highest average performance on both datasets. The black dots indicate the model performance averaged over both datasets.

models, as document layouts can be much more corpus-specific and transfer poorly to other datasets.

However, the choice of architecture also plays a role, as for the non-neural methods the image-based models still outperform the text-based models on both datasets. This is likely caused by the fact that the neural image methods have the tendency to overfit on the training data, while this is much less the case for simple baselines such as the KNN model, as this model only uses the features extracted from a pretrained VGG16 model.

4.3 Model Ensembling

For tasks that involve multiple modalities, the combination of uni-modal models often yields the best results, but there are multiple methods for combining these models, and the best variation depends on a lot factors. To investigate the best possible combination of models, both early- and late ensembling approaches are tried, and an oracle is used to have theoretical upper bounds on the performance of each of the combinations. We take an oracle in which a combined model is correct if one of the two ensembled models is correct. The score of the oracle is the maximal obtainable for the combined model. We then compare the achieved score with the maximum oracle one. As illustrated by Sharkey [23], diversity is important in successful combination of models, so the correlation between the predictions of different models is also reported.

Figure 4 shows both the Pearson correlation as well as the difference between an ensemble model and its oracle in terms of Weighted Document F1 , for all model combinations.

The Pearson correlation between two models is calculated using the page-level predictions for each model, so a binary vector where each cell is a single

Table 3. Results of the robust PSS task for the various algorithms, where a model is trained on one dataset, and tested on the other. The scores reported for OpenPSS-LONG are thus trained on OpenPSS-SHORT and tested on OpenPSS-LONG and vice-versa for OpenPSS-SHORT.Scores are reported on both page- and document level. The scores are calculated for each stream separately, and the final scores are the averages over the scores of the streams

	OpenPSS-LONG						OpenPSS-SHORT					
	Page			Document			Page			Document		
Model	P	R	F1	SQ	F1	Weighted F1	P	R	F1	SQ	F1	Weighted F1
Strong Simple Baselines												
KNN-VGG16	0.69	0.37	0.36	0.75	0.23	0.21	0.57	0.53	0.49	0.65	0.40	0.34
KNN-BERT	0.54	0.40	0.40	0.75	0.27	0.22	0.52	0.58	0.47	0.68	0.37	0.29
KNN-Ensemble	0.67	0.48	0.47	0.77	0.33	0.30	0.52	0.67	0.52	0.73	0.41	0.35
XGBoost-VGG16	0.67	0..35	0.36	0.76	0.23	0.20	0.44	0.60	0.45	0.66	0.34	0.28
XGBoost-BERT	0.65	0.31	0.33	0.61	0.19	0.16	0.69	0.44	0.48	0.62	0.38	0.31
XGBoost-Ensemble	0.66	0.36	0.33	0.72	0.22	0.19	0.67	0.50	0.50	0.65	0.41	0.33
Visual Representations												
VGG16	0.64	0.38	0.35	0.64	0.27	0.24	**0.77**	0.51	0.55	0.71	0.44	0.37
EfficientNet	0.65	0.35	0.33	0.51	0.27	0.25	0.61	0.64	0.55	0.78	0.46	0.41
Textual Representations												
BERT	0.79	0.58	0.60	0.77	0.50	0.47	0.69	0.62	0.59	0.75	0.48	0.43
TEXT-CNN	0.78	**0.64**	**0.65**	**0.85**	**0.57**	**0.53**	0.70	**0.68**	**0.62**	0.78	**0.54**	**0.49**
BERT-EfficientNet Combination												
Early Ensembling	0.76	0.41	0.41	0.74	0.31	0.28	0.67	0.67	0.60	0.79	0.52	0.47
Late Ensembling	**0.84**	0.47	0.50	0.77	0.39	0.36	0.70	0.66	**0.62**	**0.82**	0.53	0.48

page. The similarity between models of the same modality is higher than models from different modalities, indicating that the text and image models indeed pick up on different characteristics of the data.

A similar trend can be observed in the differences of the models with the oracle scores, with the combinations that have models of different modalities having the biggest room for improvement. After examining the output of the models, it was found that the predictions of the models were always very close to either zero or one, even when the model prediction was incorrect. This is a side-effect of the model training, where this behaviour is encouraged by the training objective. However, this is problematic when combining two such models, as when the models differ in classification one of them will be close to zero with the other close to one, and a linear combination will end up roughly at .5, making it difficult to make an informed decision on the input.

In an attempt to mitigate this problem, we decided to try combining the information of the models earlier in the process, before the final prediction scores. To this end, we took the outputs of the penultimate layers, and used logistic regression to combine the vectors from both models, again comparing both models from the same modality, as models from different modalities. However, this approach did not have the intended effect, and all the early-ensemble methods were outperformed by their late-ensembling counterparts.

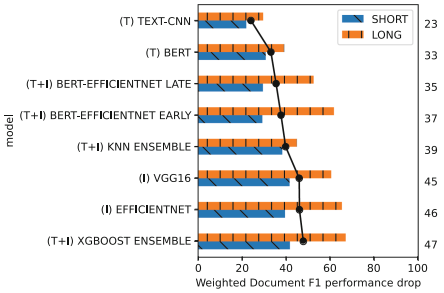

Fig. 3. Relative performance decrease in percentages for models on OpenPSS-LONG and OpenPSS-SHORT on the robust PSS experiments where a model is trained on one dataset, and tested on the other. The performance drop is calculated as $100 \cdot$ (standard score - robust score)/standard score

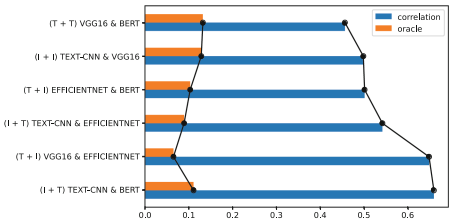

Fig. 4. Barplot of the Pearson correlation and the difference with the oracle model in Weighted Document F1 score for six model combinations, where the (maximal) oracle score is calculated with the method described in [15]

5 Relevance for Information Retrieval

Page-level classification metrics are useful when developing classifiers as they are easy to interpret, but may be misleading when considering the real task, which concerns **documents**. The document level metrics, based on the Panoptic Quality, are harder to interpret but more informative. Let us assume that we have created a search engine for the OpenPSS-LONG dataset based on the output of the best performing PSS model (the BERT-EfficientNet late ensemble). The documents ranked by the search engine are based on the partition of the stream created by the model. The scores of the model are as follows: Unweighted Document Precision: 0.82, Unweighted Document Recall: 0.82, unweighted Document F1 (RQ) 0.80 and an SQ of 0.93. Recall that for the document level scores, the true and predicted partitions are aligned and a pair (t, p) is a True Positive if the overlap between t and p is strictly larger than the non overlapping parts (formalized as $IoU(t, p) > .5$). The Segmentation Quality SQ then is the mean IoU of all True Positives.

A document precision of 0.82 means that roughly one in every 5 hits of the search engine does not correspond to a document (in the overlap is larger than non-overlap sense). That can happen in three ways. Measured on the OpenPSS-

LONG dataset the following distribution was observed: 77% of these False Positives lie inside a larger document, 14% overlaps with 2 documents and the remaining 9% with more than two.

Note that even though the document level recall is not perfect, this does not mean that some documents will not be found. Every true document D can be retrieved, but when it is a False Negative, D will be partitioned (dispersed) over other documents. Again measured on the OpenPSS-LONG dataset, we see that this dispersion looks as follows: in 93% of the cases the real document in contained within a larger predicted document, in 4% of the cases the document is dispersed over 2 documents, and in the remaining 3% the document is dispersed over more than two predicted documents.

Similarly, let us now examine the meaning of the Segmentation Quality (SQ) on the OpenPSS-LONG dataset. With a precision of .82, roughly 8 of every 10 hits is a True Positive, and thus uniquely coupled to a true document. If such a document is one or two pages long, by definition it must be a real document (because of the $IoU(t, p) > .5$ requirement), and thus the overlap is perfect. For True Positives between 3 and 10 pages long, the maximum number of non-overlapping pages can be between 2 and 9, while still being counted as a true positive. For these document lengths, the average IoU score is 0.96, and on average there is less than one non-overlapping page. For documents between 10 and 50 pages, the average IoU score is 0.92, and on average there is a mismatch of roughly 2 pages between true and predicted documents. This shows that when documents are matched correctly by the algorithm, the document boundaries are on average very closely matched with the ground truth documents.

To conclude, a search engine based on a PSS model with such good scores will probably function well. Of course, the ranking is based on the terms in the complete document, so wrong cuts can alter the ranking. However, as the difference in number of pages is rather marginal the effect will be rather small. Users may get confused if they get served a document that seemingly starts in the middle of a document, but, again as in the vast majority of cases it is only a few pages off. This may be solved by the design of the interface (e.g., the interface may show, using thumbnails, a few pages to the left and right of the starting page of the "document" in the stream).

6 Discussion and Future Work

Although we have created this benchmark with the aim of providing a platform for developing and testing for all kinds of PSS methods and approaches, the very nature of the datasets, being in Dutch, means that it is not completely language-agnostic, and that certain approaches might be limited because they would have to rely on resources for the Dutch Language, such as BERT models or word embeddings. However, this will be the case for most languages (even English to an extent), and thus we feel that is not necessarily a problem.

Possible directions for future work include the adaptation of the sequence labelling methods for the task of PSS. Although the results on the OpenPSS-LONG and OpenPSS-SHORT were not particularly strong, the method did show

potential, particularly on the OpenPSS-SHORT dataset, and the idea of incorporating information from surrounding pages seems sound. Perhaps that by adapting the LSTM approach, or by using new models such as Transformers, the limitations of the method with regards to long documents can be solved, in which case its simplicity might prove it useful as a practical PSS system.

7 Conclusion

For the first time, one can compare the best performing PSS approaches on a single large and realistic benchmark, with both page- and document-level evaluation metrics, and the possibility of evaluating the robustness of the models on out-of-distribution data. In case of the standard PSS task, the neural models that perform binary classification perform best, and ensembling BERT and Efficientnet, a combination not yet tried in the literature, achieves the best performance on both datasets. The robust task showed that the models based on textual features were the most robust to out-of-distribution data and that the image models were most susceptible to the distribution shift. A brief investigation into the different strategies for ensembling shows that the late ensembling approach achieves the best performance, and that there is still some room for improvement, based on the oracle scores.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Agam, G., Argamon, S., Frieder, O., Grossman, D., Lewis, D.: The Complex Document Image Processing (CDIP) Test Collection Project. Illinois Institute of Technology, Chicago (2006)
2. Barrow, J., Jain, R., Morariu, V., Manjunatha, V., Oard, D., Resnik, P.: A joint model for document segmentation and segment labeling. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 313–322. Association for Computational Linguistics, New York, USA (2020). https://aclanthology.org/2020.acl-main.29
3. Beeferman, D., Berger, A., Lafferty, J.: Statistical models for text segmentation. Mach. Learn. **34**(1), 177–210 (1999)
4. Braz, F.A., da Silva, N.C., Lima, J.A.S.: Leveraging effectiveness and efficiency in page stream deep segmentation. Eng. Appl. Artif. Intell. **105**, 104394 (2021)
5. Choi, F.Y.Y.: Advances in domain independent linear text segmentation. In: Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics (2000). https://aclanthology.org/A00-2004
6. Daher, H., Belaïd, A.: Document flow segmentation for business applications. In: Proceedings of the Document Recognition and Retrieval (DRR) XXI, vol. 9021, p. 90210G. International Society for Optics and Photonics (2014)

7. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 4171–4186. Association for Computational Linguistics, Minneapolis, MN, USA (2019). https://doi.org/10.18653/v1/n19-1423
8. Guha, A., Alahmadi, A., Samanta, D., Khan, M.Z., Alahmadi, A.H.: A multi-modal approach to digital document stream segmentation for title insurance domain. IEEE Access **10**, 11341–11353 (2022)
9. Hamdi, A., Coustaty, M., Joseph, A., d'Andecy, V.P., Doucet, A., Ogier, J.M.: Feature selection for document flow segmentation. In: Proceedings of the 2018 13th IAPR International Workshop on Document Analysis Systems (DAS), pp. 245–250 (2018)
10. Hernault, H., Bollegala, D., Ishizuka, M.: A sequential model for discourse segmentation. In: Gelbukh, A. (ed.) CICLing 2010. LNCS, vol. 6008, pp. 315–326. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12116-6_26
11. van Heusden, R., Kamps, J., Marx, M.: WooIR: a new open page stream segmentation dataset. In: Proceedings of the 2022 ACM SIGIR International Conference on Theory of Information Retrieval, pp. 24–33 (2022)
12. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997). https://doi.org/10.1162/neco.1997.9.8.1735
13. Kirillov, A., He, K., Girshick, R., Rother, C., Dollár, P.: Panoptic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CCVPR), pp. 9404–9413 (2019)
14. Koshorek, O., Cohen, A., Mor, N., Rotman, M., Berant, J.: Text segmentation as a supervised Learning task. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pp. 469–473. Association for Computational Linguistics, New Orleans, Louisiana (2018). https://aclanthology.org/N18-2075
15. Kuncheva, L.I.: A theoretical study on six classifier fusion strategies. IEEE Trans. Pattern Anal. Mach. Intell. **24**(2), 281–286 (2002)
16. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: Proceedings of the International Conference on Machine Learning, pp. 1188–1196. PMLR (2014)
17. Lewis, D., Agam, G., Argamon, S., Frieder, O., Grossman, D., Heard, J.: Building a test collection for complex document information processing. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 665–666. Association for Computing Machinery, New York, NY, USA (2006). https://doi.org/10.1145/1148170.1148307
18. Lukasik, M., Dadachev, B., Papineni, K., Simões, G.: Text segmentation by cross segment attention. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 4707–4716. Association for Computational Linguistics (2020). https://aclanthology.org/2020.emnlp-main.380
19. Meilender, T., Belaïd, A.: Segmentation of continuous document flow by a modified backward-forward algorithm. In: Proceedings of the Document Recognition and Retrieval (DRR) XVI, vol. 7247, p. 724705. International Society for Optics and Photonics (2009)
20. Ramachandram, D., Taylor, G.W.: Deep multimodal learning: a survey on recent advances and trends. IEEE Signal Process. Mag. **34**(6), 96–108 (2017)

21. Reynar, J.C.: An automatic method of finding topic boundaries. In: Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, pp. 331–333. Association for Computational Linguistics, Las Cruces, New Mexico, USA (1994). https://aclanthology.org/P94-1050
22. Sauvola, J., Pietikäinen, M.: Adaptive document image binarization. Pattern Recogn. **33**(2), 225–236 (2000). https://doi.org/10.1016/S0031-3203(99)00055-2
23. Sharkey, A.J.: Combining Artificial Neural Nets: Ensemble and Modular Multi-Net Systems, pp. 1–30. Springer (1999). https://doi.org/10.1007/978-1-4471-0793-4
24. Tan, M., Le, Q.: EfficientNet: rethinking model scaling for convolutional neural networks. In: Proceedings of the 36th International Conference on Machine Learning (ICML), pp. 6105–6114. PMLR (2019)
25. Wang, Y., Li, S., Yang, J.: Toward fast and accurate neural discourse segmentation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 962–967. Association for Computational Linguistics, Brussels, Belgium (2018). https://aclanthology.org/D18-1116
26. Wiedemann, G., Heyer, G.: Multi-modal page stream segmentation with convolutional neural networks. Lang. Resour. Eval. **55**(1), 127–150 (2021)
27. Zhu, G., Doermann, D.: Automatic document logo detection. In: Proceedings of the Ninth International Conference on Document Analysis and Recognition, vol. 2, pp. 864–868. IEEE (2007)
28. Zhu, G., Zheng, Y., Doermann, D., Jaeger, S.: Multi-scale structural saliency for signature detection. In: Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition (CCVPR), pp. 1–8. IEEE (2007)