WooIR: a New Open Page Stream Segmentation Dataset

Ruben van Heusden
r.j.vanheusden@uva.nl
University of Amsterdam
Amsterdam, The Netherlands

Jaap Kamps
kamps@uva.nl
University of Amsterdam
Amsterdam, The Netherlands

Maarten Marx
maartenmarx@uva.nl
University of Amsterdam
Amsterdam, The Netherlands

ABSTRACT

In this work we present *WooIR*, an open realistic benchmark for Page Stream Segmentation (PSS), the task of recovering document boundaries from aggregated *streams* of pages. Our dataset consists of over 200 streams of scanned in documents, 7K documents, 45K pages and 10M words, originating from documents released by the Dutch government in response to requests made under the Freedom of Information Act. Apart from the introduction of the dataset we perform several baseline experiments on the dataset and compare six metrics for the PSS task, in an attempt to unify the field in the usage of evaluation metrics more suited to the task. Analysis of the six metrics on the WooIR dataset shows that the dataset contains a good balance of easy and hard samples. The Panoptic Quality metric from the image segmentation field seems the most appropriate evaluation metric for the PSS task.

CCS CONCEPTS

• **Information systems** → *Evaluation of retrieval results*; **Clustering and classification**.

KEYWORDS

Page Stream Segmentation, Text classification, Clustering, Metrics, Benchmark

ACM Reference Format:
Ruben van Heusden, Jaap Kamps, and Maarten Marx. 2022. WooIR: a New Open Page Stream Segmentation Dataset. In *Proceedings of the 2022 ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR '22), July 11–12, 2022, Madrid, Spain.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3539813.3545150

1 INTRODUCTION

Having access to words and documents is a fundamental assumption underlying the field of Information Retrieval. However, there are document collections for which the unit of storage, the file, does not correspond to the unit of retrieval, the document. Typically, the documents are presented in a sequence, and the original document boundaries have to be recovered. This process is known under the name of *Page Stream Segmentation (PSS)*, and has been studied, among others, for legal [7, 25], archival [9, 14–16, 21, 25, 36], and historic [40] collections. Common to the collections are that the streams consist of a wide variety of documents, of very different lengths, usually containing scans with OCRed text, and no or few metadata available. Common to the literature on PSS is the use of non-disclosed private datasets, viewing PSS as the classification task of predicting the starting pages of documents and using evaluation metrics at the level of pages.

All approaches to PSS in some way use the idea pioneered in Hearst's TextTiling paper [19] that a drop in similarity between pages is a strong signal for a document boundary. State of the art systems use both the content (the text) and the visual form (layout, fonts, headers, images, etc) of the pages as features, take the state of the art neural architectures for text and image classification, run them in parallel, and combine the outputs or the last embedding layer to make the prediction.

For newcomers in the field, like us, it is hard to assess and compare the different approaches, because of the lack of agreed upon tasks, benchmark train and test corpora, and evaluation metrics. So we decided to fill this gap and create a publicly available benchmark, a review of proposed metrics, and a number of strong baselines. We have set up a small local competition on this benchmark and expect to have a leaderboard with the state of the art approaches available at the time of the conference.

This PSS dataset consists of documents released by the Dutch government in response to Freedom of Information (FIA) Requests. FIA requests in the Netherlands fall under the *Wet Open Overheid (WOO)* (Open Government Act), from which the dataset derives its name. In almost all cases the released documents come in the form of a non-segmented stream of documents concatenated into one (often huge) PDF file, making setting up a search engine for these FIA requests a daunting task.

The paper is organized as follows. The next section surveys related work. Section 3 lists all proposed metrics and gives uniform definitions. Sections 4 and 5 describe existing datasets and our new WooIR benchmark, and section 6 reports on a number of unsupervised baselines on this benchmark.

The dataset will become publicly available via the following url: https://irlab.science.uva.nl/resources/wooir_pss.

2 RELATED WORK

The task of splitting streams of information into consecutive and coherent blocks is a well known task that spans different modalities and has many practical applications. Think for example of detecting speaker changes in debates or the segmentation of large volumes of scanned documents in digitalization efforts.

A classic example of stream segmentation is the segmentation of a piece of text into paragraphs, or coherent pieces of text concerning the same topic, such as the segmentation of a thesis into its separate sections. One of the earliest approaches to this problem was proposed in [19], where the *TextTiling* method for splitting

ICTIR '22, July 11–12, 2022, Madrid, Spain

ACM ISBN 978-1-4503-9412-3/22/07…$15.00
https://doi.org/10.1145/3539813.3545150

paragraphs is presented. Here, the case is made for segmentation of text in paragraphs, to study phenomena such as subtopic drift, with possible extensions to discourse type data. The algorithm utilizes a lexical co-occurrence representation of sentences, and discovers boundaries by comparing the co-occurrence vectors between consecutive sentences, and placing a boundary when the similarity drops significantly. Since then, many other works dealing with paragraph level segmentation have been published, utilizing various methods [5, 12, 29, 32, 38]. Several works also use segmentation algorithms for smaller units, such as sentences or even words, for example for the segmentation of handwritten text [24, 39, 41]. The task of recovering document boundaries for (often scanned) documents is often referred to as Document Stream Segmentation (DSS) or Page Stream Segmentation (PSS) and is often a critical step in the digitalization of documents. For large collections such as historical archives or financial records from a bank, doing this manually can be a very time-consuming and expensive task. As a result, many approaches that tackle this problem exist, differing in approach from methods that use purely textual features [4, 10, 17, 28], to methods that use only visual features [1, 31] or methods that use both.

Starting with [9], methods that utilize both visual and textual features have become the standard. [9] uses bottom-up hierarchical clustering using textual and layout features; [21, 36] use visual features as fonts, margins and line separation. With the advance of neural networks, methods that do not explicitly define visual features but rather use end-to-end models from Computer Vision, possibly combined with textual models, have gained traction [7, 14, 16, 30, 40]. With the different approaches for text segmentation, different evaluation metrics are also used, varying from paper to paper, making it difficult to compare methods across papers. Moreover, most of these papers evaluate their methods on private datasets, making comparisons even harder.

A task related to Page Stream Segmentation is that of *Web Page Segmentation*, which revolves around the segmentation of web pages into coherent visual units or blocks. An overview of the task is presented by Kiesel et al. [22], which reveals many similarities between the fields of PSS and Web Page Segmentation. As with PSS, a variety of methods for segmenting the web pages exists, such as using visual features, textual features, and the usage of the DOM elements of the web page. A crucial difference with datasets used in PSS is the availability of structural information in the form of the HTML or DOM tree. Kiesel et al. argue that the usage of different incomparable evaluation metrics and datasets with lacking features hinder the progress in the task as it makes fair comparisons between metrics difficult. In their work, they present an evaluation framework based on the extended BCubed metric from Amigo et al.[2] to measure segment similarity and introduce the *Webis-WebSeg-20* dataset that contains segmentations for 8490 webpages and the first Web Page Segmentation dataset to provide all features provided by previous work in one dataset, allowing for fair comparisons of various methods that use these different features.

The segmentation task is not limited to the domains of text and images, but can also concern audio recordings, such as the detection of speaker changes in debate recordings, or detecting coherent segments in recorded lectures [13, 27]. These methods often make use of features such as intonation, prosody or structural features such as the length of pauses between speech [20].

3 METRICS

Given a consecutive input stream $S = \langle p_1, p_2 \cdots, p_n \rangle$ of length n (where p can represent any unit of information) that contains unknown boundaries, the task of *stream segmentation* is to discover the boundaries and partition the input stream.Thus, we want to partition S into k consecutive non-overlapping blocks, where k is unknown.

In fact, in certain realistic cases, the correct number of boundaries k is known, either through metadata or some other means. We will refer to this task as *k-Stream Segmentation*.

We adopt the terminology used in [9] and others. We refer to the file that contains the concatenated pages as the *stream*, and the individual parts as the *documents* which themselves consist of one ore more *pages*.

The metrics we will now survey all compare a true segmentation with a predicted or hypothesized segmentation. We can think of stream segmentation in two ways: as a *classification* problem in which we must find the start of each block, or as a *clustering* problem in which we must partition the stream into blocks of consecutive elements. Both views come with their own representation. First, we may represent a stream segmented into documents as a binary vector v whose length equals the length of the stream and in which the ones represent the starting elements. We will use t for a gold standard true segmentation and h for a predicted or hypothesized segmentation, and use $|v|$ to denote the length of v and $\sigma(v)$ to denote its sum. The dot product $t \cdot h$ counts the number of positions which have a 1 in both t and h, which is exactly the number of True Positives. The exclusive or $t \oplus h$ finds the mistakes in the prediction and $\sigma(t \oplus h)$ thus counts the number of False Positives and False Negatives.

In the clustering view, we represent a segmentation as a function p which assigns to each element in the stream a set of elements in the stream such that

- $\forall x : x \in p(x)$. Every element belongs to the set it is assigned to.
- for all x, $p(x)$ is a set of consecutive elements, in which only the first one is labelled by a 1.

Such a function is a partition and it is easy to see that for each binary vector v, there exists a unique partition p_v, and vice versa, thus the two views are interchangeable.

The metrics used in the literature can be grouped into four groups:

(1) classification metrics comparing two binary vectors;
(2) distance metrics comparing two binary vectors;
(3) classification metrics comparing the blocks in the partition;
(4) clustering metrics comparing two partitions.

We now survey these metrics group by group. In section 6, we will compare them on the WooIR set.

3.1 Classification metrics comparing binary vectors

Accuracy is an often used metric, even though the classes are usually rather imbalanced, with far fewer starting pages. The obvious alternative is precision, recall and their harmonic mean $F1$ for the starting pages. Precision and recall are easily defined using the dot product. Given two segmentations t and h of the same stream, the precision $P(t,h)$ equals $\frac{t \cdot h}{\sigma(h)}$ and the recall equals the same numerator divided by $\sigma(t)$.

Accuracy and $F1$ can be defined using the exclusive or, which indicates the false positives and negatives, and whose count equals the Hamming distance. We do that in the next subsection.

The *WindowDiff* metric introduced in [34] is a well argued improvement of the P_k measure from [6]. For a vector v, let $v[i{:}i{+}k]$ denote the subsequence of length k starting at position i. First set $D_i^k(t,h) = 1$ if $\sigma(t[i{:}i{+}k]) \neq \sigma(h[i{:}i{+}k])$, and 0 otherwise.
Then, $WindowDiff_k(t,h)$ is the mean $D_i^k(t,h)$ taken over all $1 \leq i \leq N{-}k$. This computes a sliding window over the gold standard and predicted stream, and sums the amount of times that the number of boundaries in the sliding window differ for both streams. The hyperparameter k is set to one and a half times the size of the average true document in the stream, i.e., $k = 1.5|t|/\sigma(t)$. A critique and further refinement of WindowDiff is developed in [24].

3.2 Distance metrics comparing binary vectors

The *Hamming distance* [18] between t and h equals $\sigma(t \oplus h)$, (with $\oplus$ denoting XOR) and this is the total number of errors made in h. Now we can define, given t and h, the accuracy as

$$\frac{|t| - \sigma(t \oplus h)}{|t|}$$

and the harmonic mean $F1$ as

$$\frac{t \cdot h}{t \cdot h + .5\sigma(t \oplus h)}.$$

Hamming distance counts the number of *substitutions* needed to turn one word into another. The Levenstein distance counts the minimum number of substitutions, insertions and deletions needed. These last two operations are not suited for evaluating a segmentation. For example, consider the case where the gold standard t has document separations at the even positions in the stream, and the prediction h at the odd positions. The prediction is wrong for every boundary, but by inserting a 1 at the first position and removing the last 0, the prediction can be lined up. This arguably leads to a distance score that is too low.

A lesser known fourth operation, introduced by Damerau [11], is on the other hand well suited for our task. It allows swapping two positions at the cost of 1 operation. So we can "move" a page boundary which is off by one page in one operation instead of two substitutions. We call the minimum number of swaps and substitutions needed to turn h into t the Damerau-Hamming distance between t and h. [33] propose an edit like distance measure on the blocks instead of the binary vectors, which is equivalent to the Damerau-Hamming distance.

3.3 Classification metrics comparing the blocks in the partition

The Straight Through Pass (STP) metric from [16] measures the fraction of documents in a stream that are correctly classified, and do not need any further adjustment of boundaries. So that is the *recall* of the segmentation at the level of the complete blocks. Obviously we can also define the precision and the harmonic mean at the level of blocks. We call the latter *Block F1*. Note that, when the number of blocks is known (what we called k-Stream Segmentation), precision, recall and thus $F1$ become the same measure.

This measure is very strict, as it only gives credit if the predicted document is *exactly the same* as the gold standard. Within the field of Named Entity Recognition several weaker versions have been proposed, especially when NEs tend to be long. A similar elegant weaker "partial match like" version comes from the field of image recognition and segmentation. The task is to recognize a certain object (e.g., a cancer cell) in an image *and* provide the boundaries of that object. Because one is dealing with pixels in this setting, scoring an algorithm by counting *exactly* correct bounding boxes is too strict, as being a few pixels off is usually not a problem for practical applications. The metric used to measure this is called *Panoptic Quality* (PQ), introduced in [23].

This PQ is in essence a weighted version of Block F1 in which partial matches which overlap more than half are counted as a True Positive but are weighted in the calculation of F1 by the amount of overlap. That is why we refer to it as *weighted Block F1*. The overlap between a ground truth block p_t and a predicted block p_h, is measured by their Jaccard similarity and is called *Intersection over Union* $IoU(p_h, p_t)$. A pair (p_h, p_t) is a True Positive if $IoU(p_h, p_t) > 0.5$. Note that this constraint enforces at most one True Positive pair for each true block p_t. Let TP be the set of True Positives. Then the set of False Positives FP consists of all p_h which are not part of a True Positive pair and similarly, $p_t \in FN$ iff p_t is not part of a TP pair. Now we define $F1$ as usual except that we weigh the True Positives in the numerator. Let

$$WTP = \Sigma\, \{IoU(p_h, p_t) \mid (p_h, p_t) \in TP\}.$$

Note that $0 \leq WTP \leq |TP|$, as the Jaccard similarity is bounded by 0 and 1. Now the *Weighted Block F1* is simply

$$\frac{WTP}{|TP| + .5(|FP| + |FN|)}.$$

As usual, this F1 can be equivalently defined as $2PR/(P+R)$, when we define Precision and Recall with WTP in the numerator.

3.4 Clustering metrics comparing two partitions

The survey [2] evaluates a large number of cluster quality metrics and declares the Bcubed metric [3] to be the preferred one. It is defined, given two segmentations t and h of the same stream as the mean of the BCubed $F1(e)$ scores for each element e in the stream. We use the corresponding partitions p_t and p_h to define it: for an element e in a stream

$$F1(e) = \frac{|p_h(e) \cap p_t(e)|}{|p_h(e) \cap p_t(e)| + .5 \cdot |p_h(e) \oplus p_t(e)|}$$

where $A \oplus B$ denotes the symmetric difference between the two sets A and B. Alternatively $F1(e)$ can of course be defined using the

better known formula $2P(e)R(e)/(P(e)+R(e))$, where $P(e)$ equals $\frac{|p_h(e) \cap p_t(e)|}{|p_h(e)|}$, and the recall $R(e)$ the same fraction but now with $|p_t(e)|$ as denominator.

3.5 Harmonization of Metrics

All 4 F1 metrics are comparable in range and direction, with the higher the better. $WindowDiff_k(t,h)$ is already normalized between 0 and 1, but goes in the wrong direction. So in the sequel, we will report its subtraction from 1. The Hamming-Damerau distance is normalized by dividing by the number of pages in the stream (note that by definition both streams are of equal length) [26], and we also report its subtraction from 1.

4 DATASETS

Most datasets used for evaluating Page Stream Segmentation methods are private, hindering progress in this field. We summarize some of these private datasets, and for the two publicly available datasets we perform a more detailed comparison with the WooIR dataset.

Three private datasets that have been used in recent work on PSS are *Archive22k*, the *Read Corpus* and an in-house dataset from Gordo et al.[15]. The Archive22k [40] dataset consists of 100 binders of German historical documents from between roughly 1960 and 2010, and consists of 22.741 pages. The Read corpus [30] is a corpus concerning various types of documents such as invoices and articles from journals, totalling 898 documents and 3.819 pages. In [15] an in-house dataset from the banking domain is presented, containing various document types, such as invoices, tax forms and contracts, containing 7.203 documents with roughly 70.000 pages.

Two publicly available datasets used for evaluating PSS methods are *Tobacco800* and *A.I. Lab Splitter*. Tobacco800 [25] is a subset of the Truth Tobacco Industry Documents dataset which consists of documents that became public through legal procedures against five US tobacco companies. The dataset contains a diverse set of scanned items, such as faxes, invoices an reports. The A.I. Lab Splitter is a publicly available dataset that concerns data from lawsuits and legal proceedings from courts in Brazil [7].

4.1 Dataset Comparison

We now compare the Tobacco800 and A.I. Lab Splitter datasets with the WooIR corpus on various corpus-level statistics and the method of construction and labelling.

The Tobacco800 dataset is supplied as separate pdf files, where each pdf file contains one page. Each filename has a prefix ID specifying the specific document a page belongs to, and suffixes to denote page order. To use this dataset for PSS, usually pairs of pages are sampled such that half of the pairs are pages from the same document, and the other half are pages from different documents. With PSS viewed in this way, Accuracy, and Boundary F1 are the measures that are most often used to evaluate the quality of a segmentation model.

The A.I. Lab Splitter dataset consists of 4.292 streams. The labelling has been done manually, with the help of a developed PSS tagger. Table 1 shows several statistics for the three datasets.

WooIR is the largest of the three datasets and has the lowest proportion of singleton documents. The distribution of the document lengths for all 3 sets is, as expected, heavily skewed to the right (Figure 1)

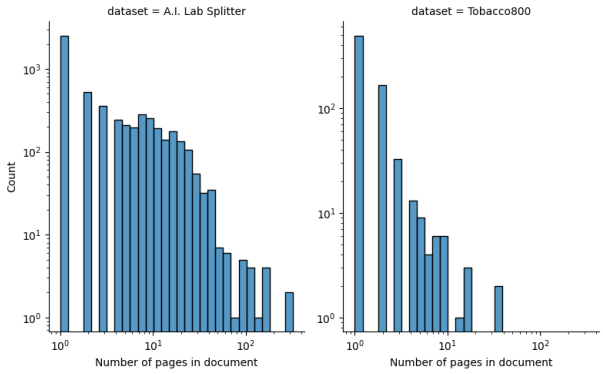


Figure 1: Histograms for the number of pages in documents for both the Tobacco800 and the A.I. Lab Splitter dataset (loglog scale).

5 WOOIR

We now describe the technical details of the WooIR dataset. It consists of two sets of streams of documents, all in Dutch, both split into a train and test set. The two test sets are held out and remain hidden, with researchers having the possibility to submit code which we will then run and return the results. Another suggested train test scenario is train on the one and test on the other.

Figures 3a and 3b show several examples of documents present in the dataset. As usual in PSS, streams contain different types of documents, with some parts being blacked out for privacy reasons.

The two train-test corpora were obtained from two rare providers of FIA-documents who made the released documents available as a zip archive instead of as a concatenated PDF. We downloaded the collection of zip archives, unzipped them, counted the number of pages of each PDF with the Linux `pdfinfo` command, and stored this as the ground truth. Similarly to [35] and [8] we concatenated the original documents into a stream. Concatenation was done in the same order as they appeared in the zip archive using `pdftk <list> cat output <concatenatedpdf>`. For every page, the text was extracted using Tesseract (version 5) OCR[1], as a command line tool, and the algorithm of Sauvola [37] was used for binarization. The exact sizes of the train and test sets for both corpora are in Table 2 and follow a $\frac{2}{3}$-$\frac{1}{3}$ random split. We investigate in Appendix A.1, A.2 and A.3 whether the two corpora are different in difficulty and whether the train and test sets are comparable.

The WooIR dataset has several unique characteristics when compared to existing datasets for Page Stream Segmentation. Not only is it publicly available, the documents in the dataset are also explicitly divided into streams, with the number of documents in the stream known, providing the opportunity for research into the *k-Stream Segmentation* task. The documents contained in one stream also belong to the same request and are expected to be topically related,

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

Table 1: General statistics of open datasets used in Page Stream Segmentation. *Streams are not present in Tobacco800. For Kurtosis, the Fisher variant was used, meaning a normal distribution has a kurtosis of 0.

Dataset	Number of streams	Number of Documents	Number of Pages	Median number of pages per document	Proportion of Singleton documents	Skew	Kurtosis
Tobacco800 [25]	N.A.*	742	1.290	1	0.67	10	123
A.I Lab splitter [7]	4.292	5.503	31.789	2	0.46	12	252
WooIR	229	7.118	44.975	2	0.32	15	350

Table 2: Basic corpus statistics of the WooIR dataset.

Corpus	Corpus 1 Train	Corpus 1 Test	Corpus 2 Train	Corpus 2 Test
Number of Streams	113	43	52	21
Number of Documents	3.914	725	2.123	356
Number of Pages	19.102	6.115	16.537	3.221
Number of Words	4.541.516	1.509.730	4.141.853	1.077.740
Vocabulary Size	155.797	83.015	189.648	55.051

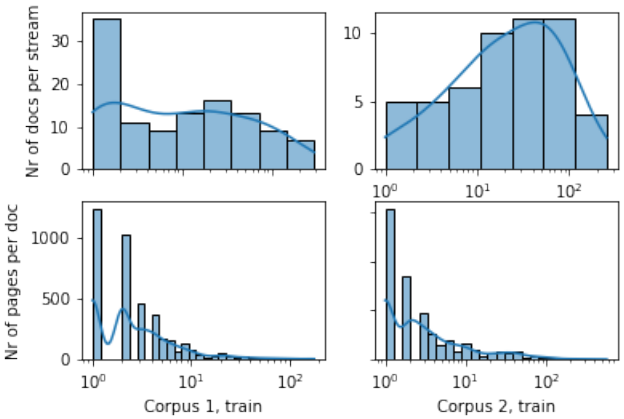


Figure 2: Population density distributions of the number of documents per file and the number of pages per document, in the train sets of both corpora. Note that the x-axis is in log scale.

adding more difficulty in boundary detection, and providing a realistic scenario for page stream segmentation. Because of the nature of the dataset and its origin, the types of documents contained in the dataset are diverse and contain among others official documents, emails, social media messages, notes, spreadsheets, and images. Almost all documents are in the Dutch Language. For PSS, the specific language is not of much influence, as long as generic state of the art tools like Transformer Language Models are publicly available. This is the case for Dutch. To the best of the authors knowledge, it is currently the largest publicly available dataset for page stream segmentation.

6 CORPUS BASELINES

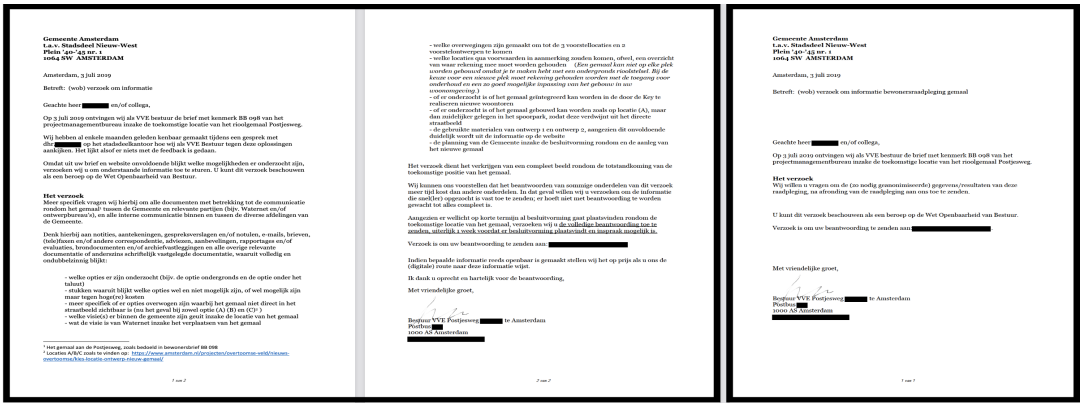
We discuss the results of running several baselines on our WooIR benchmark. None of the baselines uses learning. The two best performing baselines however rely on knowing the number of documents in a stream. We first run two extreme scenarios, putting all pages into one cluster, and considering each page a cluster. We then try fixed baselines using the corpus mean and median and more flexible baselines using the mean and median number of pages in a stream. We end with a text-only approach using agglomerative clustering. For the evaluation of the systems, we use Hamming-Damerau, WindowDiff, and the four F1 metrics. For all the baseline models presented here, the scores were reported by running the model over all data from both corpus 1 and 2. We always report the mean values over all streams.

We can conclude that all reported metrics, except for Hamming-Damerau, are appropriate for the task, that they have a good looking almost normal distribution over the WooIR dataset, and that non-learned baselines can perform quite well already. We think that the Weighted Block F1 metric has the most appropriate score distributions and makes most intuitive sense when evaluating segmentation models for an Information Retrieval task.

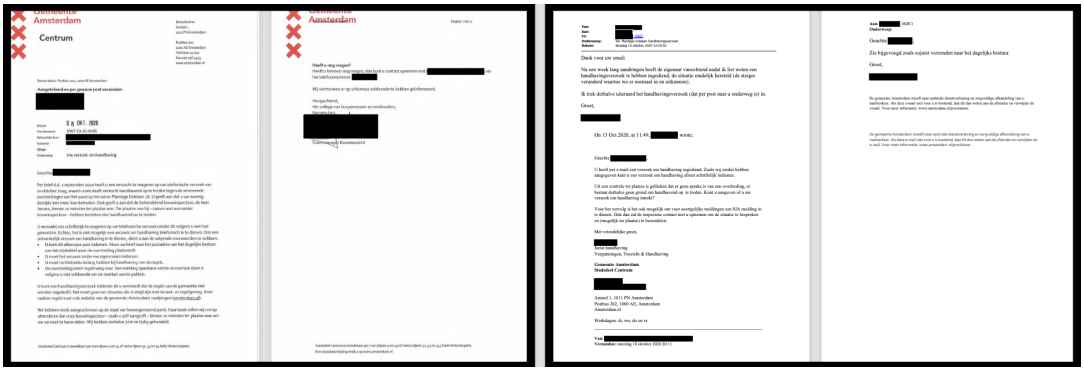
6.1 Extreme baselines

We evaluate the two extreme clustering methods, each page a cluster and only one cluster, in Table 3, containing the mean scores and in Figure 4 which shows the distribution of the metrics over the N=229 streams.

What is most striking is the large difference in the Hamming-Damerau distance for the two extremes. This can be explained by the distribution of the data. On average, only 28% of each stream consists of ones / transitions. Thus, when measuring *accuracy*, the giant cluster makes much less mistakes than the singleton clusters. For the two extreme clusterings, Hamming-Damerau is equal to Hamming, which is equal to accuracy because all elements in the prediction are either all zeros or all ones, so swapping is pointless. The BCubed scores are rather close because BCubed is the harmonic mean of precision and recall, and for both extremes these two metrics are very far apart, 'balancing' each other out, resulting in similar scores (which we believe are too high for these nonsensical extremes). The reason that WindowDiff has such a low score for the case of the singleton clusters is again because of the distribution of the document lengths. The WindowDiff metric uses



(a) Example of a stream of scanned in document from the WooIR dataset. The black boundaries indicate the individual documents.



(b) Another example of the WooIR dataset, showing headers from the municipality of Amsterdam.

Figure 3: Several examples of documents from the WooIR dataset

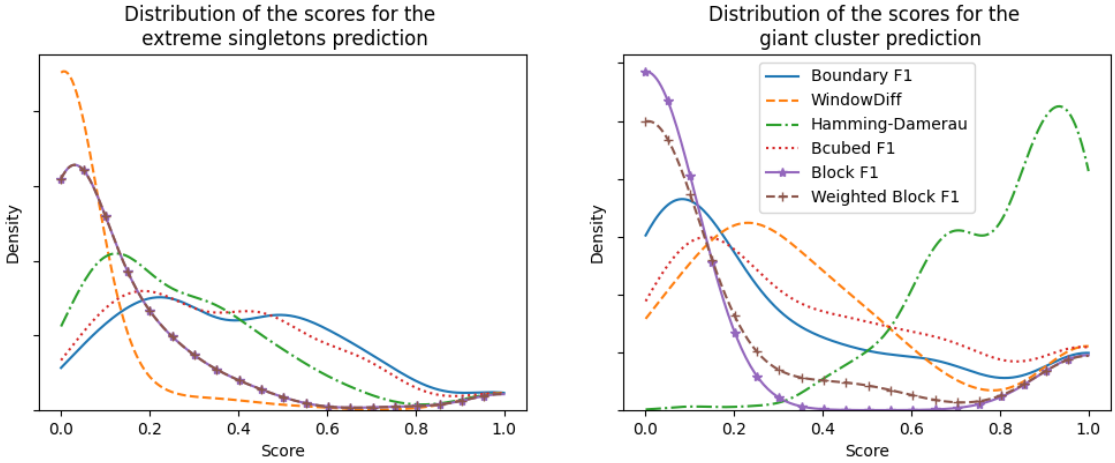


Figure 4: KDE plots of the various metrics discussed for the two extremes for the WooIR dataset. Note that for the Singletons setting, Block F1 and Weighted Block F1 are equal. (N=229)

a sliding window, and only assigns points to a model if the number of boundaries in the window is equal in both the gold standard and the prediction. Thus if there are very few boundaries and the model only predicts boundaries, the sum will almost never be equal. On the other hand, the model that never predicts boundaries receives credit more often. For the two extreme methods, both Block F1 and Weighted Block F1 have very low scores. This is because if the predicted stream has only one cluster, Block F1 will only assign

credit if the gold standard stream also only has one cluster, and Weighted Block F1 will only assign a score if there is a gold standard block with an IoU larger than 0.5. A similar story holds for the method that predicts only singletons. Note that in this case Block F1 is equal to Weighted Block F1, as each predicted block is of length 1, and thus an IoU that is larger than 0.5 must also be of length 1 and is then also an exact match.

Figure 4 shows the KDE plots of the various scores for the extreme baselines. We can see that Bcubed, Hamming-Damerau and Boundary F1 all have a distribution that is quite 'wide', with the WindowDiff, Block F1 and Weighted Block F1 metrics having a very 'peaky' distribution around 0 when examining the singletons prediction. This can in large be explained by the aforementioned reasons for the individual scores. For the case of the giant cluster predictions, things are slightly different, with the Hamming-Damerau distance having a different distribution when compared to the other metrics. This can be explained by the fact that if one giant cluster is predicted, Hamming-Damerau only penalizes boundaries in the gold standard, but gives points for correct zeros, which occur much more often in the dataset. The slight bump around a score of one for most metrics in the case of the giant cluster can be explained by the fact that there are a number of streams that only contain one document, leading to perfect scores for those streams.

Table 3: Mean scores of the two extreme baselines of one giant cluster and only singleton clusters. (N=229).

Method	Bcubed F1	Boundary F1	Hamming Damerau	Block F1	Weighted Block F1	Window Diff
singletons	0.37	0.40	0.28	0.14	0.14	0.08
giant cluster	0.40	0.32	0.81	0.14	0.19	0.38

6.2 Fixed and flexible page length baselines

We now report on more sensible baselines, using the mean and median document lengths, both for a corpus and per stream. These could be estimated from labelled data. In the case of k-stream segmentation, we know the mean document length per stream, but of course not the median. As the document lengths are right skewed, the mean is almost always larger than the median.

For the fixed document size baselines, all documents in the stream have the same length, except possibly for the remainder, which is kept as-is.

We expected that the baselines per stream would perform better than the corpus-fixed one, and that the median would be better than the mean because of the large document outliers. We can see that in the case of the stream mean segmentation, the WindowDiff and Weighted Block F1 metrics have very similar distributions, with one seeming to be a shifted version of the other. Table 4 shows that indeed the variable baselines score higher than the fixed ones, but there is no clear advantage of the median over the mean.

As with the extreme baselines, we again see of bump of the scores of most metrics around 1 in Figure 5, which can be explained by streams containing only one document or only documents of the same length, in which case taking the stream mean obviously leads to a perfect score.

Table 4: Mean scores for the corpus and stream mean and median fixed document size baselines (N=229).

	Mean & std.	Bcubed F1	Boundary F1	Hamming Damerau	Block F1	Weighted Block F1	Window Diff
Corpus mean	$\mu = 6$	0.53	0.34	0.73	0.11	0.23	0.31
Corpus median	$\mu = 2$	0.46	0.40	0.57	0.14	0.19	0.22
Stream mean	$\mu = 9.8$ $\sigma = 18.5$	**0.63**	0.46	**0.76**	0.25	0.38	**0.44**
Stream median	$\mu = 7.3$ $\sigma = 17.5$	0.60	**0.48**	0.70	**0.28**	**0.39**	0.42

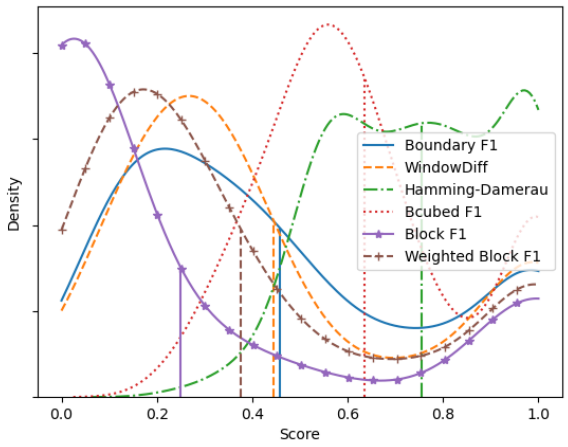


Figure 5: Metric scores for the *Stream Mean* configuration. (N=229). The vertical lines indicate the means of the various metrics.

6.3 Hierarchical Clustering baseline

In this baseline, we use the information that a stream contains some k number of documents, which we called the k stream segmentation task. We use constrained (clusters contain consecutive pages) bottom-up hierarchical clustering with cosine similarity between character ngram TF-IDF representations of pages, and single linkage, similar to [9]. The TF-IDF representations are calculated per stream, where the Document Frequency is taken over pages. The number of clusters is set equal to the known number of documents. We experimented with character ngram range and found that using 2- trough 5-grams gave the best scores. Because in this formulation of the task we supply the algorithm with the number of gold standard clusters, we remove samples that only contain one cluster as these would be trivial in this setting and unfairly inflate the score of the algorithm. This yields us a total of 205 streams.

Figure 6 shows the KDE plots for the scores obtained by hierarchical clustering. From all KDE plots this is clearly the best, as most

Table 5: Mean scores of using agglomerative clustering with 2-5 characters ngrams for $k-$ stream segmentation. (N=205)

	Bcubed F1	Boundary F1	Hamming Damerau	Block F1	Weighted Block F1	Window Diff
Hierach. Clustering	0.64	0.49	0.79	0.23	0.36	0.43

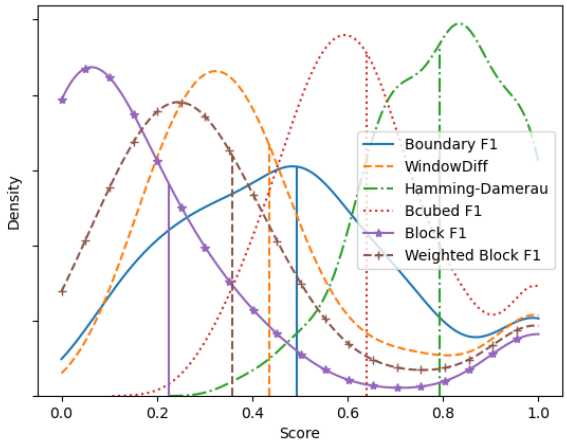


Figure 6: Metric scores for the hierarchical clustering setting (N=205). The vertical lines indicate the means of the various metrics.

metrics approach a normal distribution, and that is what we like to see for a benchmark. One thing that needs some more explanation however is the differing distribution for the Hamming-Damerau metric on the hierarchical clustering baseline when compared to the distributions of the other metrics

The Hamming-Damerau distance has much more samples with high scores than the other metrics. This can be explained by the fact that the metric does not take the severity of a mistake into account. Splitting one very large document into two documents only results in 1 mistake for the metric, while for example in BCubed, the recall of all the items is cut in half. In the case of WindowDiff, mistakes are also punished more harshly because of the sliding window. If a certain part of the stream contains no transitions but the prediction contains for example 2, then Hamming-Damerau gives two 'penalties', whereas WindowDiff might 'punish' the prediction by not giving points for multiple sliding windows, depending on the size.

7 CONCLUSION

We hope the WooIR PSS benchmark helps to bring the Page Stream Segmentation task a step further, with a clear overview of systems all evaluated in exactly the same manner, with appropiate IR-motivated metrics. Our analysis of proposed metrics and non learned baselines shows that the benchmark contains a balanced set of hard and easy train and test examples.

The two metrics at the level of pages, accuracy, here formalized as Hamming Damerau distance, and Boundary F1 present an overly optimistic view of the performance of a Page Stream Splitter. The score distributions of the other four page level metrics for the agglomarative clustering splitter (Figure 6) show that there is enough room for improvement in the dataset. We feel that Weighted Block F1, known as *Panoptic Quality* in the image segmentation literature, is the most appropiate metric for PSS when the thus splitted documents are subsequently inputted to an IR system. Future work, possibly along the lines of the desiderata of [2], has to find out whether this is indeed the case.

ACKNOWLEDGMENTS

This research was supported in part by the Netherlands Organization for Scientific Research through the ACCESS project grant CISC.CC.016 (https://www.nwo.nl/en/projects/ciscc016).

REFERENCES

[1] Onur Agin, Cagdas Ulas, Mehmet Ahat, and Can Bekar. 2015. An approach to the segmentation of multi-page document flow using binary classification. In *Sixth International Conference on Graphic and Image Processing (ICGIP 2014)*, Yulin Wang, Xudong Jiang, and David Zhang (Eds.), Vol. 9443. International Society for Optics and Photonics, SPIE, 216 – 222. https://doi.org/10.1117/12.2178778

[2] Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. 2009. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information retrieval* 12, 4 (2009), 461–486.

[3] Amit Bagga and Breck Baldwin. 1998. Entity-Based Cross-Document Coreferencing Using the Vector Space Model. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1* (Montreal, Quebec, Canada) *(ACL '98/COLING '98)*. Association for Computational Linguistics, USA, 79–85. https://doi.org/10.3115/980845.980859

[4] Joe Barrow, Rajiv Jain, Vlad Morariu, Varun Manjunatha, Douglas Oard, and Philip Resnik. 2020. A Joint Model for Document Segmentation and Segment Labeling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 313–322. https://doi.org/10.18653/v1/2020.acl-main.29

[5] Doug Beeferman, Adam Berger, and John Lafferty. 1997. Text Segmentation Using Exponential Models. In *Second Conference on Empirical Methods in Natural Language Processing*. https://aclanthology.org/W97-0304

[6] Doug Beeferman, Adam Berger, and John Lafferty. 1999. Statistical models for text segmentation. *Machine learning* 34, 1 (1999), 177–210.

[7] Fabricio Ataides Braz, Nilton Correia da Silva, and Jonathan Alis Salgado Lima. 2021. Leveraging effectiveness and efficiency in Page Stream Deep Segmentation. *Engineering Applications of Artificial Intelligence* 105 (2021), 104394.

[8] Freddy Y. Y. Choi. 2000. Advances in domain independent linear text segmentation. In *1st Meeting of the North American Chapter of the Association for Computational Linguistics*. https://aclanthology.org/A00-2004

[9] Kevyn Collins-Thompson and Radoslav Nickolov. 2002. A clustering-based algorithm for automatic document separation. In *SIGIR 2002 Workshop on Information Retrieval and OCR: From Converting Content to Grasping, Meaning*. Tampere, Finland.

[10] Hani Daher and Abdel Belaïd. 2014. Document flow segmentation for business applications. In *Document Recognition and Retrieval XXI*, Vol. 9021. International Society for Optics and Photonics, 90210G.

[11] Fred J Damerau. 1964. A technique for computer detection and correction of spelling errors. *Commun. ACM* 7, 3 (1964), 171–176.

[12] Jian Fan. 2011. Text segmentation of consumer magazines in PDF format. In *2011 International Conference on Document Analysis and Recognition*. IEEE, 794–798.

[13] Michel Galley, Kathleen R. McKeown, Eric Fosler-Lussier, and Hongyan Jing. 2003. Discourse Segmentation of Multi-Party Conversation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Sapporo, Japan, 562–569. https://doi.org/10.3115/1075096.1075167

[14] Ignazio Gallo, Lucia Noce, Alessandro Zamberletti, and Alessandro Calefati. 2016. Deep neural networks for page stream segmentation and classification. In *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*. IEEE, 1–7.

[15] Albert Gordo, Marçal Rusinol, Dimosthenis Karatzas, and Andrew D Bagdanov. 2013. Document classification and page stream segmentation for digital mailroom

applications. In *2013 12th International Conference on Document Analysis and Recognition*. IEEE, 621–625.
[16] Abhijit Guha, Abdulrahman Alahmadi, Debabrata Samanta, Mohammad Zubair Khan, and Ahmed H Alahmadi. 2022. A Multi-Modal Approach to Digital Document Stream Segmentation for Title Insurance Domain. *IEEE Access* 10 (2022), 11341–11353.
[17] Ahmed Hamdi, Mickaël Coustaty, Aurelie Joseph, Vincent Poulain d'Andecy, Antoine Doucet, and Jean-Marc Ogier. 2018. Feature selection for document flow segmentation. In *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*. IEEE, 245–250.
[18] Richard W Hamming. 1950. Error detecting and error correcting codes. *The Bell system technical journal* 29, 2 (1950), 147–160.
[19] Marti A Hearst. 1997. Text tiling: Segmenting text into multi-paragraph subtopic passages. *Computational linguistics* 23, 1 (1997), 33–64.
[20] Julia Hirschberg and Christine H Nakatani. 1998. Acoustic indicators of topic segmentation. In *Fifth International Conference on Spoken Language Processing*.
[21] Romain Karpinski and Abdel Belaïd. 2016. Combination of structural and factual descriptors for document stream segmentation. In *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*. IEEE, 221–226.
[22] Johannes Kiesel, Florian Kneist, Lars Meyer, Kristof Komlossy, Benno Stein, and Martin Potthast. 2020. Web Page Segmentation Revisited: Evaluation Framework and Dataset. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management* (Virtual Event, Ireland) *(CIKM '20)*. Association for Computing Machinery, New York, NY, USA, 3047–3054. https://doi.org/10.1145/3340531.3412782
[23] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. 2019. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9404–9413.
[24] Sylvain Lamprier, Tassadit Amghar, Bernard Levrat, and Frederic Saubion. 2007. ClassStruggle: A Clustering Based Text Segmentation. In *Proceedings of the 2007 ACM Symposium on Applied Computing* (Seoul, Korea) *(SAC '07)*. Association for Computing Machinery, New York, NY, USA, 600–604. https://doi.org/10.1145/1244002.1244140
[25] D. Lewis, G. Agam, S. Argamon, O. Frieder, D. Grossman, and J. Heard. 2006. 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* (Seattle, Washington, USA) *(SIGIR '06)*. Association for Computing Machinery, New York, NY, USA, 665–666. https://doi.org/10.1145/1148170.1148307
[26] Stan Z. Li and Anil Jain (Eds.). 2009. *Hamming Distance*. Springer US, Boston, MA, 668–668. https://doi.org/10.1007/978-0-387-73003-5_956
[27] Igor Igor Mikhailovich Malioutov. 2006. *Minimum cut model for spoken lecture segmentation*. Ph.D. Dissertation. Massachusetts Institute of Technology.
[28] Th Meilender and Abdel Belaïd. 2009. Segmentation of continuous document flow by a modified backward-forward algorithm. In *Document Recognition and Retrieval XVI*, Vol. 7247. International Society for Optics and Photonics, 724705.
[29] Hemant Misra, François Yvon, Olivier Cappé, and Joemon Jose. 2011. Text segmentation: A topic modeling perspective. *Information Processing & Management* 47, 4 (2011), 528–544.
[30] Chems Neche, Yolande Belaïd, and Abdel Belaïd. 2020. Use of language models for document stream segmentation. In *International Conference on Pattern Recognition Applications and Methods*.
[31] Sofia Ares Oliveira, Benoit Seguin, and Frederic Kaplan. 2018. dhSegment: A generic deep-learning approach for document segmentation. In *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, 7–12.
[32] Shashank Paliwal and Vikram Pudi. 2012. Investigating Usage of Text Segmentation and Inter-passage Similarities to Improve Text Document Clustering. In *Machine Learning and Data Mining in Pattern Recognition*, Petra Perner (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 555–565.
[33] Patrick Pantel and Dekang Lin. 2002. Efficiently clustering documents with committees. In *Pacific Rim International Conference on Artificial Intelligence*. Springer, 424–433.
[34] Lev Pevzner and Marti A Hearst. 2002. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics* 28, 1 (2002), 19–36.
[35] Jeffrey C. Reynar. 1994. An Automatic Method of Finding Topic Boundaries. In *32nd Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Las Cruces, New Mexico, USA, 331–333. https://doi.org/10.3115/981732.981783
[36] Marçal Rusinol, Volkmar Frinken, Dimosthenis Karatzas, Andrew D Bagdanov, and Josep Lladós. 2014. Multimodal page classification in administrative document image streams. *International Journal on Document Analysis and Recognition (IJDAR)* 17, 4 (2014), 331–341.
[37] Jaakko Sauvola and Matti Pietikäinen. 2000. Adaptive document image binarization. *Pattern recognition* 33, 2 (2000), 225–236.
[38] Fei Song, William M. Darling, Adnan Duric, and Fred W. Kroon. 2011. An Iterative Approach to Text Segmentation. In *Proceedings of the 33rd European Conference on Advances in Information Retrieval* (Dublin, Ireland) *(ECIR'11)*. Springer-Verlag, Berlin, Heidelberg, 629–640.
[39] Yi Sun, Timothy S Butler, Alex Shafarenko, Rod Adams, Martin Loomes, and Neil Davey. 2007. Word segmentation of handwritten text using supervised classification techniques. *Applied Soft Computing* 7, 1 (2007), 71–88.
[40] Gregor Wiedemann and Gerhard Heyer. 2018. Page Stream Segmentation with Convolutional Neural Nets Combining Textual and Visual Features. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. European Language Resources Association (ELRA), Miyazaki, Japan. https://aclanthology.org/L18-1581
[41] Jingbo Zhu, Muhua Zhu, Huizhen Wang, and Benjamin K. Tsou. 2009. Aspect-Based Sentence Segmentation for Sentiment Summarization. In *Proceedings of the 1st International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion* (Hong Kong, China) *(TSA '09)*. Association for Computing Machinery, New York, NY, USA, 65–72. https://doi.org/10.1145/1651461.1651474

A APPENDIX

A.1 KDE plots for number of documents per stream

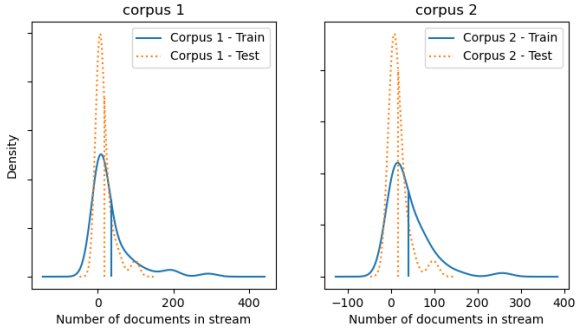


Figure 7: Kernel Density Estimation plots for the corpus 1 and corpus 2 respectively, with the distributions for train and test overlayed on each other. ($N_{corpus1-train}$ = 113, $N_{corpus1-test}$ = 43, $N_{corpus2-train}$ = 52, $N_{corpus2-test}$ = 21)

A.2 K-stream hierarchical clustering for all subcorpora

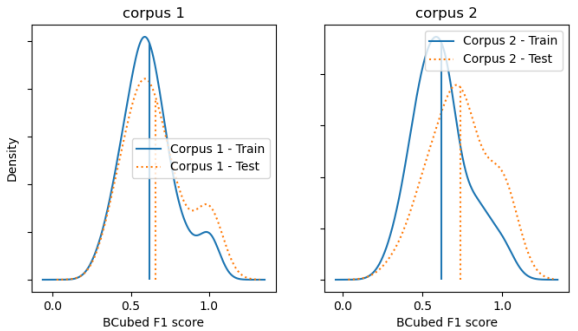


Figure 8: Kernel Density Estimation plots BCubed F1 scores for the K-Stream hierarchical segmentation algorithm reported for both corpus 1 and 2 separately. ($N_{corpus1-train}$ = 92, $N_{corpus1-test}$ = 43, $N_{corpus2-train}$ = 49, $N_{corpus2-test}$ = 21)

Figures 7 and 8 show the KDE plots for the number of documents per stream and the BCubed F1 score for the train and test portions of both corpus 1 and corpus 2. Both figures show that the differences in distributions between the train and test portions of both corpora are small and Figure 8 shows that there do not appear to be any major differences in BCubed F1 score distribution between the corpora, indicating that the train test splits for both corpora are representative of the train portions.

A.3 Kolmogoroc-Smirnov tests

Number of documents in stream	
Compared Sets	P value
Train 1 - Test 1	0.42
Train 2 - Test 2	0.051
All 1 - All 2	0.03
BCubed F1 Score	
Compared Sets	P value
Train 1 - Test 1	0.57
Train 2 - Test 2	0.006
All 1 - All 2	0.47

Table 6: P values for the Kolmogoroc-Smirnov tests for the data presented in Figure 7 and 8. Test comparing the train and test portions of both corpus 1 and corpus 2 against each other were conducted, as well as comparing all of corpus 1 with all of corpus 2.

In order to perform a more detailed analysis on the significance of the differences between train and test portions and corpus 1 and corpus 2 as a whole, we performed two-sample Kolmogoroc-Smirnov tests on the distributions shown in Figures 7 and 8 and reported the scores in Table 6. We take a p value smaller than 0.05 to reject the hypothesis that both samples came from the same underlying distribution. From the table we can see that although for corpus 1 the number of documents per stream and the BCubed F1 scores for the train and test set are not significantly different, this is not the case for corpus 2. For corpus 2, both the number of documents per stream and the BCubed F1 scores for train and test are significantly different, although the sample sizes for both sets are considerably smaller than their corpus 1 counterparts. Finally, although the distribution of the number of documents per stream is significantly different between corpus 1 and corpus 2, the BCubed F1 scores are not.