PaperClip: Automated Dossier Reorganizing

Wessel Stoop[1,2], Iris Hendrickx[2] and Tom van Ees[1]
[1]*Davinci Products, Amsterdam, The Netherlands*
[2]*Centre for Language & Speech Technology / Centre for Language Studies, Radboud University, Nijmegen, The Netherlands*
{*wstoop, tvanees*}*@davincigroep.nl, i.hendrickx@let.ru.nl*

Keywords: Dossier Reorganizing, Text Classification, Pagenumber Classification, Customer Document Processing.

Abstract: We investigate the creation of a robust algorithm for document identification and page ordering in a digital mail room in the banking sector. PaperClip is a system that takes dossiers containing pages of various documents as input, and returns multiple files that contain all the pages of one document in the correct order. PaperClip performs (1) document type classification and (2) page number classification on each page, and then (3) merges the results. We experimented with various algorithms and methods for these three steps and we performed an elaborate evaluation to measure different aspects of the methods. The best performing setup achieved a cut F-score of 86% and a V-measure of 0.91%. This performance is sufficient to fulfill business needs of the banking sector.

1 INTRODUCTION

There is a clear need for office document automation that offers a real-time and secure processing, especially in the banking sector where processing of streams of scanned and digitized administrative documents form an important work flow. Automatic document analysis is the field that deals with the different steps in the document processing pipeline, from the incoming documents to document specific treatment. A mayor part of this field focuses on digital image analysis (Marinai, 2008). Here we focus on the problems occurring at the start of the document processing pipeline and instead of doing image analysis, we concentrate on textual content analysis of OCR-ed versions of the scanned documents.

Financial institutions frequently receive or store all customer related business documents in one dossier: a single pdf file for each customer. A customer may for example make scans of his/her driver's license, bank statements and salary slips in one go, and save the results as a single file. In this way all documents are conveniently bundled and stored together. Further detailed automatic content analysis is typically done with document type specific information extraction programs. Such information extraction programs work with single documents and this implies that these dossiers need to be split into separate documents for further processing. To make the matter more complicated, the individual pages of documents in the dossier may be shuffled, or pages of multiple documents might be mixed. In this study we focus on the development of a robust algorithm for document identification and page ordering.

Our goal is to find the best algorithm to automatically reorganize a dossier and that cuts up dossiers in a set of smaller documents such that each file only consists of pages belonging to same true document. This entails (1) document type classification of individual pages, (2) page number detection and (3) combining the page and document type information to decide where to split the dossier in separate documents.

In this study we aim to discover what the optimal methods for each of these steps are. We evaluated various algorithms and methods for these three steps and the resulting output set of documents. We performed an elaborate evaluation to measure different aspects of the methods. As we will explain in more detail in section 3.3, we evaluate the output set both on cutting into separate files and on grouping those pages together that actually belong to the same true document. The optimal combination of these steps was implemented in one system that we named PaperClip. This application has already been used in a number of real life situations such as a customer loan office.

1.1 Related Work

The area of document analysis in office automation covers a broad range from filtering and rearranging

Stoop, W., Hendrickx, I. and Ees, T.
PaperClip: Automated Dossier Reorganizing.
DOI: 10.5220/0006195904710478
In *Proceedings of the 6th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2017)*, pages 471-478
ISBN: 978-989-758-222-6

incoming documents (the focus of the current study) up to detailed structural analysis for document understanding as was done for example in (Klink and Kieninger, 2001). We refer to Marinai (2008) for a general overview and introduction on document analysis and recognition.

Many recent approaches for the classification of identity documents rely on graphical instead of textual information (Chen et al., 2012a,b; Infantino et al., 2014; Kumar et al., 2014; Simon et al., 2015). This has the advantage of not being dependent on the idiom used in the documents (Infantino et al., 2014), and can also rely on graphical clues like layout and images (Kumar et al., 2014). Other studies combine both approaches such as Gordo et al. (2012) who present a study on document type classification on a large in-house office documents data set. They compare visual and textual information and show that a textual bag-of-words representation outperforms a system trained on visual features.

Rusiñol et al. (2014) focuses on page classification of administrative documents in a banking work flow, very similar to our work. The authors present a system that combines visual and textual information from the documents. The image classification focuses on pixel density in multiple scales, the textual information uses a tf*idf weighing scheme and a compressed semantic LSA (Deerwester et al., 1990) representation. Several combinations of the two cues are evaluated.

Verberne et al. (2010) try to classify patent documents according to the International Patent Classification system. They do so by feeding the textual content in various ways to the algorithms Winnow, Support Vector Machines and Naive Bayes. They report that SVM and Winnow are roughly equal in terms of performance, and slightly better than Naive Bayes. However, Matwin and Sazonova (2012) concluded that Naive Bayes performed better when classifying medical abstracts in an otherwise comparable experiment using SVM and Naive Bayes.

For page number classification, most related work focuses on page numbers as part of an overall document structure analysis like for example (Klink and Kieninger, 2001) who aimed to detect page-number blocks in the visual page lay-out.

An overall method for automatically splitting files into smaller files using document type classification of digital images by using a combination of classification and/or rules was patented by Schmidtler et al. (2014). In a highly comparable fashion to our problem, Agin et al. (2015) concentrated on the problem of splitting an incoming document flow into separate documents in a banking scenario. Interestingly, they focus on predicting the transition points between two separate files and use a SVM classifier to learn whether two pages are a continuation of the same file or not. They focus solely on visual features in this study, and actually suggest that for further improvements, adding textual content would be a good future research direction.

2 METHOD

2.1 System Overview

We designed a system called PaperClip that takes a file containing multiple documents as input, and writes one or more output files to a separate folder. Ideally these output files all contain exactly one document. PaperClip uses a rather straightforward approach that emerged from various pilot experiments. Firstly, the individual pages of the dossier are classified on a) document type and b) page number. These single page classifications are used in a second phase where the system aims to split the dossier into smaller coherent files. These steps are described in more detail in sections 2.2 and 2.3 respectively.

2.2 Single Page Classification

In PaperClip's first stage, the text on each page is classified twice: one time for document type (for all classes, see table 1), and one time for page number. With *page number* the position of the page in the original true document is meant. The page number is often indicated by a small number in the top or bottom part of the page, but not always (for example, there is no '2' on scans of the backside of an identity card).

The text of each page was extracted from the source document by OCR tool Abbyy Recognition Server 3.1. The resulting XML files contain information of all identified characters, their positions, and how they are grouped in lines, among other things. This information was converted into a list of words and punctuation, which was offered as simple bag of words frequency vectors to the classifiers. The casing was kept as it was, and no additional information was added.

For both classification rounds, we tested three algorithms that are known to work well on text classification tasks (Sebastiani, 2002):

1. Gaussian Naive Bayes, as implemented in the Scikit Learn Python package (Pedregosa et al., 2011).

2. Linear Support Vector Machines, as implemented in the Scikit Learn Python package (Pedregosa et al., 2011).
3. Balanced Winnow, as implemented in the Linguistic Classification System (Koster et al., 2003).

2.3 Page Sequence Processing

Classifying the individual pages gives us a most likely document type and page number for each page. In the next phase, we aim to process the sequence of pages in the dossier and split them properly in smaller files that match the original true document boundaries.

For this sequence processing step, we need to represent the individual page number and document type information in an efficient and truthful way. Such representation is not trivial, and a simple IOB (Inside, Outside, Beginning) tagging (Tjong Kim Sang and Veenstra, 1999) cannot handle the problem of wrong page ordering and consequent sequence changes. We decided to represent the output sequence as a string of the format 'a1a2b2b1a3', where each page is encoded by two characters: a letter and a number. The number represents the page number of the page in the original document (so not the dossier), and each letter identifies a unique document. Thus a file with output string 'a1a2b2b1a3' contains two documents: one with three pages in the correct order (found on page 1, 2 and 5 of the dossier), and one with two pages in reverse order (found on page 3 and 4 of the dossier).

To convert these classification results to actual decisions, we tested three algorithms:

1. SameDocumentUnlessSamePageNumber (SDUSPN). This algorithm walks through the pages one by one, and glues every page to all pages previously encountered with the same document type. Only when the pagenumber of a particular page was previously encountered, it assumes a new document has started.
2. SameDocumentUnlessUnexpectedPageNumber (SDUUPN). This algorithm is similar to the previous one, but requires an extra condition to be fulfilled before it assumes that a page belongs to a document previously identified: it should also have a pagenumber that is adjacent to the last page encountered of this document.
3. SameDocumentUnlessGap (SDUG). This algorithm walks through the pages one by one, and glues each page to the previous page as long as the previous page has a different pagenumber, but the same document type.

All algorithms can also be run evaluating the pages in reverse order. This leads to six possible ways to merge the classification results into an output string. As an example, say we have a dossier consisting of 7 pages with the following most likely document types and pagenumbers: [(SalarySlip, page 1), (BankStatement, page 1), (BankStatement, page 2), (BankStatement, page 3), (BankStatement, page 2), (SalarySlip, page 2),(SalarySlip, page 7)]. The results of the six merging options (all of which could be correct, depending on which pager actually were from the same document):

1. SameDocumentUnlessSamePageNumber, forward. a1b1b2b3c2a2a7
2. SameDocumentUnlessGap, forward. a1b1b2b3c2d2d7
3. SameDocumentUnlessUnexpectedPageNumber, forward. a1b1b2b3c2a2d7
4. SameDocumentUnlessSamePageNumber, backwards. a1b1b2c3c2a2a7
5. SameDocumentUnlessGap, backwards. a1b1b2c3c2d2d7
6. SameDocumentUnlessUnexpectedPageNumber, backward. b1c1c2d3d2b2a7

3 EXPERIMENTS

We randomly split the data set in two parts, 80% (40 dossiers) was used for training the PaperClip system and a held-out set of 20% (10 dossiers, 268 pages) was used for evaluation. We ran experiments for all combinations of the three different classifiers for both modules.

3.1 Baseline

As a baseline, we used two alternatives to PaperClip:

1. CutEverywhere, which simply assumes each dossier only consists of one page documents.
2. SimpleDocumentTypeClassification, which has access to information about the document types, and assumes that all adjacent pages of the same document type belong to the same document.

3.2 Dataset

Unfortunately, publicly available data sets for document type classification (Marinai, 2008) do not resemble the type of administrative documents that we work with. Therefore we use a data set of real life dossiers acquired from a Dutch anonymous company that provides consumer loans. These dossiers consist

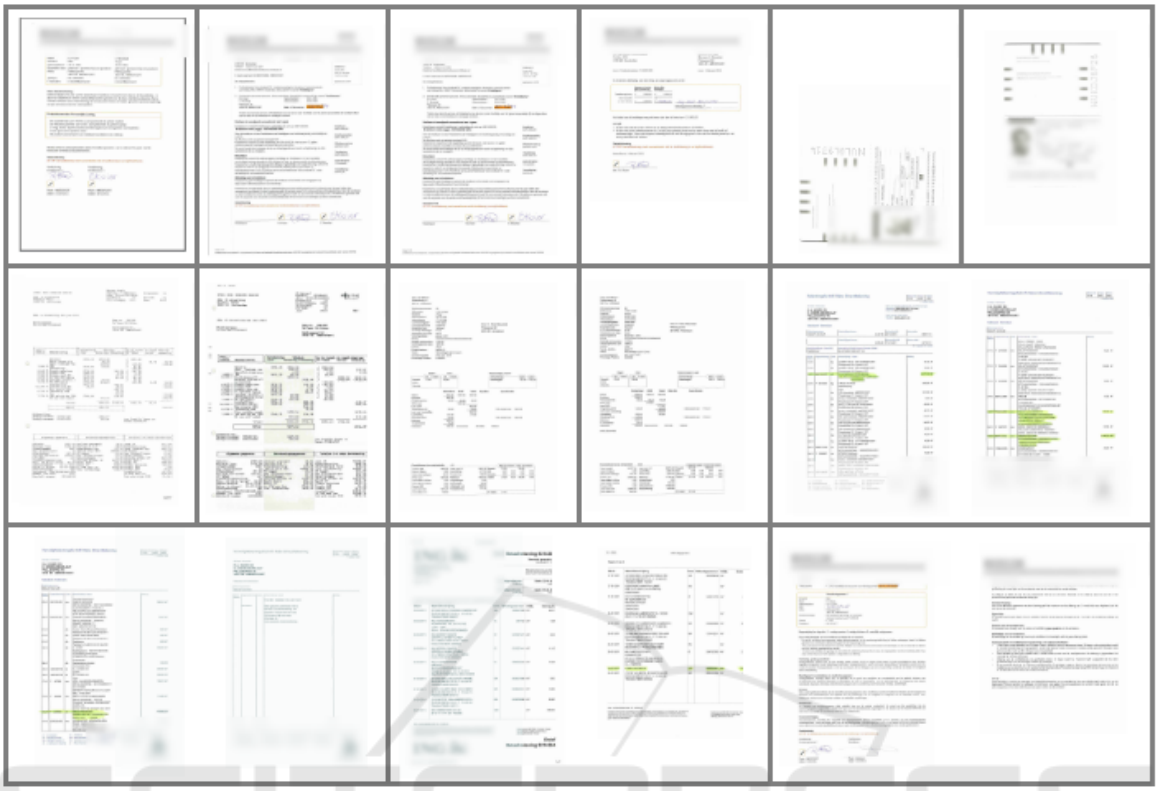

Figure 1: Example dossier, vertical lines indicate a new document (unreadable on purpose).

of 30 pages on average, and contain various documents of 1 or more pages. In total, there were 1522 pages, bundled in 50 dossiers. Figure 1 shows a full example dossier. Following Simon et al. (2015), who also work with privacy sensitive information, we made the example too small to read on purpose, and blurred out graphical information.

All pages were manually labeled for document type and page number (classes were the numbers 1-17). In case a particular document did not match any of the predefined list of document types, it was labeled *Miscellaneous*.

This category was later analyzed again to identify recurring document types not covered by the predefined list, which resulted in the extra document types *Email*, *Insurance* and *HousingContract*. Because the annotation task was simple and straightforward, only one annotator was considered necessary. Table 1 gives an overview of all 18 document type labels that were used, and of how many pages these documents had in our data set.

Table 1: For each document type: the mean number of pages, in what percentage of dossiers it occurs, and the mean number of occurrences in a single dossier when present.

Document type label	# of pages	% of dossiers	# per dossier
BalanceBill	1.04	0.54	1.67
BankStatement	2.23	0.96	3.25
CoverLetter	1	0.30	1.40
DivorceSettlement	4.67	0.08	1.50
Email	1.67	0.18	1.33
EmployerDeclaration	1	0.02	1
EmptyPage	1.0	0.02	5
HousingRentContract	1.37	0.34	4.24
LoanInfo	1.42	0.68	7.94
Mortgage	1.0	0.02	2
Miscellaneous	1.46	0.56	4.39
ID	1.29	0.92	1.74
Insurance	1.33	0.34	1.29
PensionInformation	1.75	0.14	4.86
PensionStatement	1.4	0.02	5.0
ResearchedPersonalInfo	1.5	0.02	2
SalarySlip	1.02	0.96	2.87
WorkContract	1.93	0.36	1.72

3.3 Evaluation

We first report on the quality measures of the two classification rounds on single pages. However, these results do not fully capture to what extent the final goal is met. This is because it is possible, as indeed happens in practice, that the correct final decisions are made despite mistakes by the classifiers. For example, the classifiers might label the first and second page of an identity card as the fourth and fifth page of a salary slip; although this is completely wrong, the two pages

will still be cut and glued together correctly.

At a first glance, it seems a reasonable option to evaluate the predicted PaperClip output page sequence by comparing against a true page-number sequence representation. For example, we can match the true constructed sequence 'a1a2b1b2' to the predicted string 'a1a2b1c1' and count the differences. However, such evaluation method often does not suffice for two reasons:

1. A small mistake in the beginning can shift the entire output string. For example, if the second page in 'a1a2b1b2c1c2' (a2) is incorrectly recognized as a new document, this leads to an output string where all following letters are different: 'a1b2c1c2d1d2'.
2. In some cases mistakes have no effect on the desired outcome of correctly reorganizing a dossier: when the second page in 'a1a2b1b2' is incorrectly recognized as a third page (so the system output is 'a1a3b1b2'), the dossier will still be organized correctly.

Therefore, we mainly focus on two groups of more sophisticated evaluation metrics:

1. *Cut precision*, *cut recall* and *cut F-score*. These metrics consider the transition between each of the pages in the dossier If PaperClip correctly makes a cut at a particular transition (that is, there also is a cut according to the annotation), this is considered a True Positive. Incorrect cuts by PaperClip are considered as False Positives, missed cuts are counted as False Negatives, and if PaperClip does not make a cut where there indeed should not be one, this is considered a True Negative. On the basis of the resulting confusion matrix, the *cut precision* and *cut recall* can be calculated, which represent whether all predicted cuts are correct, and whether all cuts that should have been made are actually done, respectively. The harmonic mean of the cut precision and cut recall is called the *cut F-score*. The *cut F-score* thus represents to what extent PaperClip makes cuts on the correct places.
2. *Homogeneity*, *completeness*, *V-measure* and *Adjusted Rand Index*. The *cut F-score* does not capture whether particular groups of pages are correctly identified to be of the same document. To fill this gap, we use evaluation metrics that are traditionally used for clustering algorithms. Each document in a dossier is viewed as a cluster and we evaluated to what extent the clusters detected by PaperClip match the true clusters. *Homogeneity* entails to what extent pages that were detected to be from a single document are indeed part of a single true document. *Completeness* entails to what extent pages truly belonging to the same document were also detected to be from the same document. The harmonic mean of homogeneity and completeness is called the *V-measure* (Rosenberg and Hirschberg, 2007).

   Because there are many single page documents in the data set, and the algorithm is biased towards cutting between pages, we expect that many of PaperClip's decisions will be correct by chance. The *V-measure* does not control for this. For this reason, the *Adjusted Rand Index* (ARI) is also included, which checks for each pair of pages whether it is from the same document, compares this to corresponding pair of pages from the annotated dossier, and controls for chance. The *V-measure* and the *Adjusted Rand Index* thus represents whether pages that are from the same document are also identified to be from the same document.

4 RESULTS

We first report on the phase of individual page classification with the three different machine learning algorithms. The best performing algorithm was then used for the page sequence processing.

4.1 Individual Page Classification

We report on the performance of the individual classifiers on the held-out test set of 268 pages. The performance of recognizing document types is summarized in table 2, the performance of the pages number classification for the first 5 pages is in table 3. Furthermore, we report both the macro and the macro average. For the macro average we first compute the average score per document type, sum these averages and then divide by the number of types, for micro average we compute the score per page and divide by the total number of pages.

We observe a lot of variation between document types, ranging from an F-score of 0.18 for the BalanceBill by Naive Bayes to an F-score of 1 for the PensionStatement by Winnow. Winnow outperforms the other two classifiers for most document types and on average, and the same is true for the page numbers.

4.2 Page Sequence Processing

We show the baseline results on page sequence processing in table 4. The baseline algorithm CutEvery-

Table 2: Precision (p), recall (r) and F-score (f) of individual document type classification on the held out set. Classes present in the training material but not in the test set are omitted.

Document type	#	Naive Bayes			SVM			Winnow		
BalanceBill	8	p 0.50	r 0.25	f 0.33	p 0.67	r 0.50	f 0.57	p 0.56	r 0.62	f 0.59
BankStatement	53	p 0.88	r 1.00	f 0.94	p 0.88	r 0.98	f 0.93	p 0.93	r 1.00	f 0.96
CoverLetter	3	p 1.00	r 0.67	f 0.8	p 1.00	r 0.67	f 0.8	p 1.00	r 0.67	f 0.8
DivorceSettlement	10	p 1.00	r 0.10	f 0.18	p 0.78	r 0.7	f 0.74	p 1.00	r 0.70	f 0.82
Email	2	p 0.00	r 0.00	f 0.00	p 1.00	r 1.00	f 1.00	p 1.00	r 1.00	f 1.00
HousingRentContract	20	p 0.59	r 0.80	f 0.68	p 0.62	r 0.65	f 0.63	p 0.85	r 0.85	f 0.85
ID	19	p 0.93	r 0.68	f 0.79	p 1.00	r 0.63	f 0.77	p 0.94	r 0.89	f 0.92
Insurance	4	p 1.00	r 0.25	f 0.40	p 1.00	r 0.25	f 0.40	p 0.50	r 0.25	f 0.33
LoanInfo	58	p 0.68	r 0.81	f 0.74	p 0.79	r 0.98	f 0.88	p 0.85	r 1.00	f 0.92
Miscellaneous	31	p 0.53	r 0.58	f 0.55	p 0.59	r 0.32	f 0.42	p 0.62	r 0.48	f 0.55
PensionStatement	5	p 0.57	r 0.8	f 0.67	p 1.00	r 0.40	f 0.57	p 1.00	r 1.00	f 1.00
SalarySlip	26	p 1.00	r 0.92	f 0.96	p 1.00	r 0.96	f 0.98	p 0.96	r 0.96	f 0.96
WorkContract	10	p 1.00	r 0.50	f 0.67	p 0.67	r 0.6	f 0.63	p 0.88	r 0.7	f 0.78
Average (micro)		p 0.74	r 0.74	f 0.74	p 0.77	r 0.77	f 0.77	p 0.86	r 0.86	f 0.86
Average (macro)		p 0.74	r 0.57	f 0.59	p 0.85	r 0.66	f 0.72	p 0.85	r 0.78	f 0.81

Table 3: Precision (p), recall (r) and F-score (f) of individual page number classification on the held out set.

Page Number	#	Naive Bayes			SVM			Winnow		
1	153	p 0.68	r 0.97	f 0.8	p 0.81	r 0.93	f 0.87	p 0.81	r 0.95	f 0.88
2	37	p 0.75	r 0.32	f 0.45	p 0.42	r 0.51	f 0.46	p 0.59	r 0.59	f 0.59
3	20	p 0.89	r 0.4	f 0.55	p 0.5	r 0.45	f 0.47	p 0.52	r 0.55	f 0.54
4	11	p 1.00	r 0.27	f 0.43	p 1.00	r 0.36	f 0.53	p 0.88	r 0.64	f 0.74
5	7	p 1.00	r 0.14	f 0.25	p 0.67	r 0.29	f 0.40	p 1.00	r 0.29	f 0.44
Average (micro)		p 0.69	r 0.69	f 0.69	p 0.71	r 0.71	f 0.71	p 0.74	r 0.74	f 0.74
Average (macro)		p 0.86	r 0.42	f 0.5	p 0.68	r 0.51	f 0.55	p 0.76	r 0.604	f 0.64

Table 4: Performance of baselines (PaperClip alternatives) Cut Everywhere (CU) and Simple Document Type Classification (SDTC).

	CU	SDTC
Cut precision	62.09	86.51
Cut recall	100.0	50.32
Cut F-score	75.59	59.4
Homogeneity	1	0.61
Completeness	0.78	0.93
V-measure	0.87	0.72
ARI	0.0	0.3

where performs well (due to the many 1-page documents). The relatively high results of CutEverywhere can be explained by the high recall that follows from the algorithm: because it always makes a cut, it will by default cover all cuts that needed to be made at the cost of making unnecessary cuts, resulting in a lower precision.

We compare the baseline method results shown in table 4 against the results of the six different sequence processing methods described in section 2.3 using the best individual page classifier (Winnow for both tasks) in table 5.

Focusing on the Cut F-score we see that Winnow outperforms the baselines regardless of the merging algorithm used. This means that adding Winnow for both document type and page number classification leads to more useful cuts than simply splitting the dossiers in single pages or to only use document type classification.

We see a similar pattern for internal relations: CutEveryWhere performs good at Homogeneity, but is outperformed by Winnow because Winnow is able to give a more balanced result. This leads to a higher completeness, V-measure and ARI. Furthermore, we see that homogeneity consistently has a higher score than completeness, indicating that pages that PaperClip groups together are indeed from the same document, but that there sometimes are even more pages. Another thing that might strike the eye is that the Adjusted Rand Index is lower than homogeneity, completeness and the V-measure because it corrects for chance. This shows that part of the correct decisions by PaperClip can indeed be attributed to chance, but also that PaperClip clearly performs better than chance. In a purely chance based system, the ARI would be 0, as demonstrated by the CutEveryWhere baseline.

As for the merging algorithms, we see that the for-

Table 5: Merging algorithm results with Winnow for both classification tasks, F(orward) and B(ackward).

Algorithm	SDUSPN		SDUUPN		SDUG	
Direction	F	B	F	B	F	B
Cut precision	79.37	72.92	75.1	74.41	77.53	73.38
Cut recall	97.0	91.2	98.5	97.14	97.83	91.2
Cut F-score	85.99	79.67	84.13	82.81	85.19	80.0
Homogeneity	0.98	0.95	0.99	0.98	0.99	0.96
Completeness	0.86	0.84	0.85	0.84	0.87	0.84
V-measure	0.91	0.89	0.91	0.9	0.92	0.89
ARI	0.5	0.35	0.44	0.41	0.52	0.33

ward versions consistently outperform the backward versions. The differences between the three forward versions turn out to be minimal, in particular for the balancing metrics Cut F-score. Any differences are related to whether false positives of false negatives are considered more problematic: SDUSPN has a higher precision, but a lower recall, SDUUPN has the reverse, and SDUG is in between. For the internal relations, we note that SDUUPN has a lower ARI, but we believe that the rest of the differences are too small to be meaningful.

An advantage of Winnow is that it allows to look 'inside' the model to see what terms it has identified as good predictors for a particular class. Interestingly, for the page number classification, these are not only page numbers; it also uses other general page indications and words typically appearing on particular pages of several document types such as 'voorschot' (deposit), 'openbare' (available) and 'geschieden' (archaic form of 'to happen').

The reason for the latter phenomenon is that identity documents and most contract(-like) document types use partly fixed formats on designated pages, where whole paragraphs of text are copied from a standard model. Furthermore, we observe that even in more free form documents the same content is handled on roughly the same places in the document.

5 DISCUSSION

The following three issues have been identified as main challenges for the current implementation of PaperClip:

1. Identity documents are often misclassified by PaperClip, because (1) there is often multiple such documents in the dossier and (2) they are smaller than a typical scanned page, which encourages customers to scan two documents on a single page. Multidocument pages are not yet supported. The expectation is that the only way to extract multiple documents from a single page is with a more graphical approach.
2. Over-reliance on single page documents. As shown by table 1, by far most documents consist of only one page. This means that the safest choice is to always make a cut; most classifiers would learn this quickly. This is indeed what we see in table 5, where recall is consistently higher than precision, indicating that PaperClip 'cuts when in doubt'. A possible solution could be to apply down-sampling, but we expect that down-sampling would be accompanied by a significant performance drop, at least partly resulting from the decrease in the size of the training corpus.
3. Classification problems caused by unseen document types in the category labeled *Miscellaneous*. Classification of this diverse category was problematic for all classifiers, as learning from training examples for this category provided little information about unseen instances. A solution for this problem could be to include another algorithm into the system, which would identify whether a particular document is similar enough to the training material (and otherwise label it as 'new document type'). We believe, however, that the most practical solution would be to simply increase the amount of training documents, thereby increasing the chance that a particular document type is present.

6 CONCLUSION

Our goal was to find the best algorithm to automatically reorganize a dossier. PaperClip does this by performing (1) document type classification and (2) pagenumber classification on each page, and then (3) merges the results to make decisions on where to make cuts and reorder pages the original input dossier. For the first two steps the text classification algorithms Naive Bayes, SVM and Winnow were evaluated, with Winnow outperforming the other two on both tasks. The results of Winnow were then applied with six variations of merging algorithms, all of which performed better than the baseline algo-

rithms CutEveryWhere and SimpleDocumentTypeClassification. The forward versions of these merging algorithm performed better than the backward variations. The best performing setup achieved a cut F-score of 86% and a V-measure of 0.91%. This is a satisfactory result to fulfill business needs of the banking sector and PaperClip is already being used in real life.

REFERENCES

Agin, O., Ulas, C., Ahat, M., and Bekar, C. (2015). An approach to the segmentation of multi-page document flow using binary classification. In *Proceedings of the Sixth International Conference on Graphic and Image Processing (ICGIP 2014)*, pages 944311–944311. International Society for Optics and Photonics.

Chen, F., Girgensohn, A., Cooper, M., Lu, Y., and Filby, G. (2012a). Genre identification for office document search and browsing. *International Journal on Document Analysis and Recognition (IJDAR)*, 15(3):167–182. note Iris: not-so relevant: text-based features but not OCR-features.

Chen, S., He, Y., Sun, J., and Naoi, S. (2012b). Structured document classification by matching local salient features. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR)*, pages 653–656.

Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., and Harshman, R. A. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

Gordo, A., Perronnin, F., and Valveny, E. (2012). Document classification using multiple views. In *Document Analysis Systems (DAS), 2012 10th IAPR International Workshop on*, pages 33–37. IEEE.

Infantino, I., Maniscalco, U., Stabile, D., and Vella, F. (2014). A fully visual based business document classification system. In *Proceedings of the Science and Information Conference (SAI), 2014*, pages 339–344. IEEE.

Klink, S. and Kieninger, T. (2001). Rule-based document structure understanding with a fuzzy combination of layout and textual features. *International Journal on Document Analysis and Recognition*, 4(1):18–26.

Koster, C. H. A., Seutter, M., and Beney, J. (2003). *Perspectives of System Informatics: 5th International Andrei Ershov Memorial Conference, PSI 2003, Akademgorodok, Novosibirsk, Russia, July 9-12, 2003. Revised Papers*, chapter Multi-classification of Patent Applications with Winnow, pages 546–555. Springer Berlin Heidelberg, Berlin, Heidelberg.

Kumar, J., Ye, P., and Doermann, D. (2014). Structural similarity for document image classification and retrieval. *Pattern Recognition Letters*, 43:119 – 126. {ICPR2012} Awarded Papers.

Marinai, S. (2008). Introduction to document analysis and recognition. In *Machine learning in document analysis and recognition*, pages 1–20. Springer.

Matwin, S. and Sazonova, V. (2012). Direct comparison between support vector machine and multinomial naive bayes algorithms for medical abstract classification. *JAMIA*, 19(5):917.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Rosenberg, A. and Hirschberg, J. (2007). V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure. In *Proceedings of EMNLP-CoNLL*, volume 7, pages 410–420.

Rusiñol, M., Frinken, V., Karatzas, D., Bagdanov, A. D., and Lladós, J. (2014). Multimodal page classification in administrative document image streams. *International Journal on Document Analysis and Recognition (IJDAR)*, 17(4):331–341.

Schmidtler, M. A., Texeira, S. S., Harris, C. K., Samat, S., Borrey, R., and Macciola, A. (2014). Automatic document separation. US Patent 8,693,043.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47.

Simon, M., Rodner, E., and Denzler, J. (2015). Fine-grained classification of identity document types with only one example. In *Machine Vision Applications (MVA), 2015 14th IAPR International Conference on*, pages 126–129. IEEE.

Tjong Kim Sang, E. and Veenstra, J. (1999). Representing text chunks. In *Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics*, pages 173–179. Association for Computational Linguistics.

Verberne, S., Vogel, M., D'hondt, E., et al. (2010). Patent classification experiments with the linguistic classification system lcs. In *CLEF (Notebook Papers/LABs/Workshops)*.