

Improving Public Access to Government Documents

Ruben van Heusden

Improving Public Access to Government Documents

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. ir. P.P.C.C. Verbeek
ten overstaan van een door het College voor Promoties ingestelde
commissie,
in het openbaar te verdedigen in de Aula der Universiteit
op donderdag 11 december 2025, te 14:00 uur

door

Ruben Jacob van Heusden

geboren te Haarlem

Promotiecommissie

<i>Promotor:</i>	dr. ir. J. Kamps	Universiteit van Amsterdam
<i>Copromotor:</i>	dr. M.J. Marx	Universiteit van Amsterdam
<i>Overige leden:</i>	prof. dr. A. Doucet	La Rochelle Université
	dr. E.H. Saaman	Nationaal Archief
	prof. dr. K.J.P.F.M. Jeurgens	Universiteit van Amsterdam
	dr. L. Stork	Universiteit van Amsterdam
	prof. dr. M. de Rijke	Universiteit van Amsterdam

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

The research was carried out at the Information Retrieval Lab at the University of Amsterdam, with support from NWO under project number CISC.CC.016.

Copyright © 2025 Ruben van Heusden, Amsterdam, The Netherlands
Cover by Ruben van Heusden
Printed by Ridderprint, The Netherlands

ISBN: 978-90-821695-4-6

Acknowledgements

When I started my PhD journey almost five years ago, little did I know about both the challenges I would face (especially doing a PhD during the COVID-19 pandemic) as well as the incredible experiences I would have, visiting many great conferences and countries, and meeting people without whom this whole PhD would not have been possible.

Firstly, I would like to sincerely thank my promoter Jaap Kamps, who has provided great guidance during these years, and who has helped me in defining the direction of my research, especially in the beginning, when I had little clue about what to do and what I would like to work on. Secondly, I would like to thank my co-promoter, Maarten Marx, for being a great daily supervisor, helping me navigate the sometimes difficult world of academic research, and being patient with my lack of an eye for finer details. I fondly remember the many interesting discussions we have had in your office over the years, working on papers together, and your often much needed help with my continuing struggle with writing mathematical proofs. One of the perks of being your only PhD student has been that you were almost never too busy to help me with any questions I had, and for that I am very grateful. Apart from your academic guidance, I think back to our frequent discussions about music, and you introducing me to bands I had never before heard of.

I would also like to take this opportunity to thank Antoine, Erik, Charles, Lise and Maarten for being part of my defense committee, and for taking the time and effort to read my dissertation.

Although doing a PhD can often be quite a lonely journey, I have had the great pleasure of having an amazing group of colleagues and friends, who have helped me tremendously throughout these five years, both in- and outside of the lab. I would like to specifically thank Vera for being a great friend and paronymph, sharing much of our everyday PhD struggles, as well as creating our very own band and recording our own songs. I remember the many fun conversations we had at Taalcafe, hanging around the center of Madrid in the middle of the night during SIGIR, or going for a walk when we were tired of doing research. I also want to thank Sam, for being a mentor of sorts and guiding me on various PhD-related topics, I always knew who to ask when I was having questions about using the cluster, evaluation forms or conferences. Apart from this, playing guitar together with you provided me with a sometimes much needed break from research, and although probably should have prepared better, running a 5K together with you was a fun experience.

Thilina, I think now is the time we can no longer pretend we still have a lot of time left, and that we can put off writing our thesis. Apart from distracting other people from working with our frequent coffee breaks, I fondly remember the many times we played boardgames at your place, and the many dinners and lunches that you organized, as well as the many late nights at Oerknal during the Friday drinks, making the lab a more social place, and showing me that there really is more besides just work. Speaking of this, I want to thank Vera, Sam, Philipp, Jasmin, Syrenna and Natalie for participating in our jam sessions, singing and generally enjoying music together.

I also want to thank the past and present members of the IRLab/ ILPS, who have all

been amazing colleagues, and together created the amazing atmosphere in the lab, with the group outings, reading groups, and the many social activities that kept us all sane during COVID. A big thanks to Alessio, Ali, Ali, Amir, Ana, Andrew, Antonis, Arezoo, Barrie, Chang, Chuan, Clara, Clemencia, Dan, David, Dylan, Evangelos, Gabriel, Georgios, Harrie, Hongyi Ilias, Ilya, Jasmin, Jiahuan, Jie, Jin, Jingfen, Jingwei, Julien, Kidist, Maartje, Maarten de Rijke, Mahsa, Maria, Mariya, Marzieh, Maurits, Maryam, Ming, Mohammad, Mounia, Mozhdah, Negin, Olivier, Pablo, Philipp, Pooya, Romain, Roxana, Ruqing, Sami, Sebastian, Shaojie, Shashank, Shubha, Simon, Songgaojun, Svetlana, Thilina, Thong, Vaishali, Weiya, Xinyi, Yangjun, Yibin, Yifei, Yongkang, Yougang, Yuanna, Yuyue, Zahra, Zihan, and Ziming.

I would also like to thank the Bachelor and Master students that I have had the pleasure to supervise, and who have greatly inspired me with their enthusiasm for their projects. Thank you Anne, Aron, Femke, Gerda, Jenny, Kaj, Maik, Miguel, and Pepijn!

Apart from my friends from the lab, I would like to thank Syrenna for being a great friend throughout these years. I have known you for most of my life, and I cannot remember a time when we did not talk regularly, be it in person or online. I know you have had to listen to your fair share of my rants about my PhD, but you were always the voice of reason that helped me through some difficult times. Also a big thanks to Michel, I have known you since high school, when we bonded over music, and went on a trip to Belgium with our guitars. Although we don't live close anymore these days, our yearly holiday tradition has endured, and what better way to take a break from a PhD then to bike across the country on an electric tandem bike?

Finally I would like to thank my sister, my mom, my dad and the rest of my family for supporting me throughout my PhD, without your continuing support this would not have been possible, and I feel deeply grateful for having been provided with the ability to pursue my dreams, and you stimulating my curiosity from an early age.

Ruben
Haarlem, October 2025

1	Introduction	1
1.1	Research Outline and Questions	2
1.1.1	Document Processing Technology (for FOIA Documents) . .	2
1.1.2	Evaluation of Extreme Document Segmentation and Clustering	4
1.2	Main Contributions	6
1.3	Thesis Overview	7
1.4	Origins	8
I	Document Processing Technology for FOIA Documents	13
2	OpenPSS: An Open Page Stream Segmentation Benchmark	15
2.1	Introduction	16
2.2	Related Work	17
2.2.1	Page Stream Segmentation	17
2.2.2	Other Datasets	18
2.3	Method	18
2.3.1	Datasets	18
2.3.2	PSS Variants	20
2.3.3	Models	20
2.3.4	Model Ensembling	22
2.3.5	Evaluation	22
2.4	Results	23
2.4.1	Standard Page Stream Segmentation Task	23
2.4.2	Robust Page Stream Segmentation Task	25
2.4.3	Model Ensembling	26
2.5	Relevance for Information Retrieval	27
2.6	Discussion & Future Work	28
2.7	Conclusion	29
3	Redacted Text Detection Using Neural Image Segmentation Methods	31
3.1	Introduction	32
3.2	Related Work	34
3.2.1	Detection of Redacted Text	34
3.2.2	Neural Image Segmentation	34
3.3	Methodology	35
3.3.1	Data	35
3.3.2	Models	37
3.3.3	Mask Post-Processing	39
3.3.4	Computational Resources	40
3.3.5	Evaluation	40
3.3.6	Code Availability	40
3.4	Results	41
3.4.1	Redacted Text Detection	41

3.4.2	Adding Pages Without Redactions	43
3.4.3	Influence of the Number of Training Samples	45
3.4.4	Post-processing Model Output	45
3.5	Discussion & Future Work	46
3.6	Conclusion	47
4	A Collection of FAIR Dutch Freedom of Information Act Documents	49
4.1	Background & Summary	49
4.2	Methods	52
4.2.1	Dataset Structure	52
4.2.2	Dataset Overview	52
4.2.3	Dataset Collection	54
4.2.4	FAIRification	54
4.3	Data Records	58
4.4	Technical Validation	59
4.5	Usage Notes	62
4.6	Code Availability	62
4.7	Conclusion	62
II	Evaluation of Extreme Document Segmentation and Cluster-	65
ing		
5	Elements Like Me: BCubed Revisited	67
5.1	Introduction	68
5.2	BCubed Revisited	68
5.2.1	A New Name	70
5.2.2	First Impression of the Differences	70
5.3	Evaluation	70
5.3.1	ZeroScore constraint	71
5.3.2	ELM Behaves Well on Degenerate Clusterings	72
5.3.3	ELM Can Produce Different Rankings Compared to BCubed .	72
5.3.4	ELM vs BCubed on synthetic data	73
5.3.5	ELM Vs. BCubed on Real Data	74
5.3.6	ELM Satisfies the Constraints of Amigó et al	77
5.4	BCubed in the Literature	80
5.5	Discussion	81
5.6	Conclusion	81
6	A Sharper Definition of Alignment for Panoptic Quality	83
6.1	Introduction	83
6.2	Theoretical results	86
6.2.1	Every fair alignment satisfies $(\theta^{\&})$	90
6.3	Empirical Evaluation	91
6.4	Related Work	95
6.5	Discussion & Future Work	96

6.6	Conclusion	97
7	Text Segmentation Metrics: A Survey	99
7.1	Introduction	99
7.2	Evaluation criteria	101
7.2.1	The Space of Text Segmentation Metrics	102
7.3	Formal definitions of text segmentation metrics	104
7.3.1	Segment-level metrics	104
7.3.2	Element-level metrics	106
7.3.3	Metrics based on a sliding window	108
7.3.4	Metrics on an example segmentation	110
7.4	Metric Evaluation	113
7.4.1	Helpfulness	113
7.4.2	Distinctiveness	119
7.4.3	Meaningfulness	120
7.4.4	Scrutiny	121
7.4.5	Overview of Metric Qualities	123
7.5	Overlapping segmentations	124
7.6	Conclusion	125
8	Conclusions	127
8.1	Main Findings	128
8.1.1	Document Processing Technology (for FOIA Documents)	128
8.1.2	Evaluation of Extreme Document Segmentation and Clustering	129
8.2	Future Work	131
	Bibliography	133
	Summary	143
	Samenvatting	145

1

Introduction

Every day, governments at all levels produce massive quantities of information in many different file types and formats. Freedom of Information Act (FOIA) legislation requires that much of this information be released either passively (on request) or actively, by the respective agencies. This type of legislation allows citizens and journalists alike to gain insight into the specific procedures of governments, obtain information relevant to them, and, in a broader sense, to contribute to the functioning of a democratic society. With the advent of digital technologies in recent years, most of this information is now published online on government websites or other platforms.

For example, in the Netherlands, we have a Freedom of Information Act (called *Wet Openbaarheid Bestuur*, *WOB*) since 1991, and a more general Open Government Act (called *Wet Open Overheid*, *WOO*) since 2022. The implication is that citizens and journalists can request information from their government, and that the government has to provide the requested documents unless there are exemption grounds (such as privacy or national security). In addition, the government has to proactively release documents of an increasing number of document types, in particular those reflecting decisions of the government of interest to citizens. Obviously, just the publication of this information by the government is a major effort, due to the scale on which documents are created by the government, and the number of different government agencies involved. At the time of writing, the first and most important step, the publication of these open government documents, has been implemented, but little thought has gone into what happens after publication.

Collecting all the released open government documents in a single search engine, allowing for the finding and (re)using of open government documents across different branches of government, is a challenging task.¹ This requires harvesting open government documents from over a thousand suppliers in the Netherlands alone. Due to a lack of centralized creation and collection, the released documents often fail to meet the FAIR data guidelines outlined by Wilkinson et al. [122]. First of all, even though regulation exists regarding the type of documents that should be released, there are no

¹In light of the increased digital publication of documents, the Platform Open Overheid (PLOOI) was set up, with the intent of creating a single centralized platform containing the documents from all Dutch governmental agencies, with the ability for users to search these document collections easily (<https://www.overheid.nl/help/plooi>). Unfortunately, problems with the acquisition of the data and the creation of the search engine have meant that the project was eventually scrapped in 2023 (<https://www.digitaleoverheid.nl/nieuws/plooi-gaat-verder-in-eenvoudigere-variant/>).

set standards on the metadata associated with these records. As such, there exists a large variety in metadata across different organizations. Moreover, the quality of the documents themselves is often poor, with for example little to no machine-readable text, and often lacking information that allows for example visually impaired people to interact with the data. All of these things combined mean that, although potentially a valuable resource for citizens and research alike, in its current form, much of the data published online by government agencies has very limited usefulness.

In an ideal scenario, we could address these problems at the source, creating standardized procedures for the publication of documents and their metadata, and having quality control mechanisms in place to guarantee high-quality data, interoperable between different suppliers. However, the sheer number of suppliers involved means that this is not a short- or midterm solution, and that repairing document inadequacies after publication is currently the most viable solution.

To investigate an immediate solution, this thesis addresses the task of improving the quality of government documents by developing and evaluating post-hoc processing techniques that help improve the *Findability*, *Accessibility*, *Interoperability*, and *Re-usability* (FAIRness) of these documents. This is a key step towards realizing the potential of open government information to journalists and citizens alike, thereby improving government transparency and democratic control.

1.1 Research Outline and Questions

The core focus of this thesis is on *developing* and *evaluating* methods for improving the quality of released government information, such that these documents adhere to FAIR data principles. As such, the work is focused on the following main research question:

Main Research Question *Can we develop enabling technology to improve FOIA document quality at scale, and how do we evaluate the quality of these extreme document clustering and segmentation tasks?*

Specifically, the thesis deals with page stream segmentation of complex dossiers into individual documents and redacted text detection to benefit downstream page segmentation and OCR output. Both these document engineering tasks are extreme clustering tasks, requiring us to critically revisit existing evaluation measures for page stream- and image segmentation. The thesis consists of two parts, exploring the development and evaluation of these methods in Parts I and II, respectively.

1.1.1 Document Processing Technology (for FOIA Documents)

The first part of the thesis is focused on developing techniques to improve document quality, with the aim of making the documents more usable for researchers and citizens alike. As such, we pose the following objective:

Key Objective 1 *To develop effective document processing technology for FOIA documents.*

We focus on developing automatic processing techniques for two common document processing tasks: page stream segmentation and the automated detection of redacted text in FOIA documents. Having surveyed the efficacy of different techniques for both tasks, we then use the best-performing models for these tasks to aid in the construction of a large-scale dataset of Dutch FOIA documents that adheres to the FAIR data principles outlined before. This leads to the following three chapters.

Chapter 2 is concerned with the task of Page Stream Segmentation (PSS), and how Machine Learning methods can be used to automate part of this task. The task of PSS deals with the segmentation of streams of (digitized) pages into the original source documents. This concatenation of documents is a frequent artifact of document scanning, and it can have adverse effects on downstream tasks that rely on cleanly separated documents as input. Many different approaches have been suggested throughout the literature, often differing in the modality used (images, text or both) or the view of the task (binary- or sequence classification). In an attempt to obtain a comprehensive overview of the performance of current models and approaches, this chapter poses the following research question:

RQ1 *What is the efficacy of methods from Machine Learning for the task of Page Stream Segmentation?*

In order to answer this question, we construct a benchmark consisting of two large public datasets originating from Dutch FOIA documents and evaluate four different groups of algorithms. We evaluate these groups both in an in-distribution setup, where a model is trained and tested on data from the same dataset, as well as an out-of-distribution setup, where one dataset is used for training and another is used for testing. In addition, we explore several strategies for model combination in an attempt to increase model performance. Our experiments show that a neural network trained using a page-classification scheme performs best, where a multimodal approach is best for an in-distribution setup, and a method based on only textual data is best for an out-of-distribution scenario.

Chapter 3 deals with another part of document digitization, namely the detection of redacted text. Many documents, and in particular documents released through FOIA legislation, may contain sensitive information that has to be redacted before release in order to protect the privacy of the concerned individuals or organizations. Being able to detect the locations of these redactions within a page automatically has several uses, including aiding text-to-speech tools in handling redacted pieces of text or in providing an accurate estimation of the amount of redaction in a document collection.

The detection of redacted text has not been investigated extensively in the literature, but the task can be seen as a particular instance of image segmentation, for which a variety of methods have been proposed. In order to provide a preliminary overview of the usage of neural image segmentation methods for this task, we pose the following research question:

RQ2 *What is the efficacy of neural image segmentation methods for the large-scale detection of redacted text?*

We answer this research question by constructing an annotated dataset of documents containing redactions and comparing a rule-based model using morphological operations with two state-of-the-art models for image segmentation. In order to accurately mimic the conditions of a real-world scenario, we also evaluate the models on documents that do not contain any redactions. Our experiments show that both neural methods significantly outperform the rule-based method in their ability to detect redactions, as well as producing fewer false positives when presented with documents containing no redactions.

Chapter 4 is a culmination of the previous two chapters, and discusses the creation of the Woogle dataset, a large-scale collection of Dutch FOIA documents collected from over a thousand suppliers in the Netherlands. The dataset includes the results from the best performing methods for PSS and the detection of redacted text from Chapters 2 and 3, as well as the usage of other off-the-shelf automatic techniques for enhancing the document and metadata quality of the collection. Since the creation of such a large collection of FOIA documents is an involved task with many moving parts, and this project aimed to evaluate how feasible this goal of having a central platform with these documents really is, we posed the following research question:

RQ3 *What lessons can be learned from a Living Lab of FOIA documents?*

This chapter provides an overview of the dataset created and the results of some of the algorithms developed in the previous two chapters on this large collection of documents. Although the results in the paper show part of the outcomes, it is also clear that the digitization of a large corpus is very intricate, and that many problems have to be addressed before such a corpus is ready for usage in more sophisticated applications, such as usage with Large Language Models or Information Retrieval systems.

This part developed a range of document processing technology for FOIA documents and demonstrated their effectiveness on the Woogle dataset. This makes an important contribution to improving access to FOIA documents at scale, and how relatively simple, off-the-shelf tools can be used to effectively address problems with released FOI documents.

1.1.2 Evaluation of Extreme Document Segmentation and Clustering

Both page stream segmentation and the detection of redacted text we covered in Part I can be seen as clustering tasks, and as such metrics from this domain could be used to measure their performance. The second part of the thesis goes into more detail about the evaluation aspect of extreme document segmentation and clustering tasks, and poses the following objective:

Key Objective 2 *To Evaluate Extreme Document Segmentation and Clustering Tasks.*

However, there is a large set of possible metrics, based on different principles, making it difficult to select a single metric for evaluation, and there is little agreement in the

literature on the best evaluation metrics to use for different tasks. We address these issues by investigating two commonly-used clustering methods: BCubed for text clustering, and PQ for image clustering/segmentation, and conclude with an evaluation of these and other metrics on the task of text segmentation. This leads to the following three chapters.

Chapter 5 starts with an in-depth investigation of the BCubed metric [5], an element-wise clustering metric that has seen widespread usage. Although intuitive at first glance, there are several desirable properties that the metric lacks, such as its inability to assign a zero score to clustering predictions that are completely incorrect. As such, we pose the following research question:

RQ4 *Can the BCubed metric be repaired in such a way that its shortcomings are addressed while still maintaining its desirable theoretical properties?*

We answer this question in the affirmative by proposing *ELM*, a variation on the BCubed metric, and by comparing the two metrics both theoretically and empirically. We first compare both metrics on a synthetic dataset, comparing the distributions of scores, and evaluate their differences when used to rank a set of hypothesized clusterings against a true clustering. We continue with a comparison of both metrics on a real-world text segmentation dataset, and finish up with a theoretical proof that our proposed repair still satisfies the clustering metric constraints posed by Amigo et al. [3]. The experimental results show that the ELM metric closely matches the distribution of scores attained by BCubed but has the ability to generate different systems rankings, while maintaining the desirable properties of the original formulation.

Chapter 6 is concerned with the Panoptic Quality metric, a segmentation metric that was originally developed for Computer Vision tasks by Kirillov et al. [60] but that, due to its versatility, is also applicable to clustering problems in general. The metric provides a means of evaluating object detection methods by imposing a partial bijection over the clusters in the true and hypothesized sets, by requiring the number of common pixels in the overlap of two clusters to be larger than the number of unmatched pixels of both clusters. Although the choice of threshold seems natural, it is not the only possible way to create one-to-one mappings between true and hypothesis segments, and a more general mapping may exist that is better suited for measurements. As such, we propose the following research question:

RQ5 *Is there an objective mathematical criterion for defining a matching function that ensures a one-on-one mapping between two sets of (non-overlapping) clusters?*

We answer this research question in the affirmative by theoretically proving that such a matching definition indeed exists, and that it is most general. We conduct a set of experiments on three image segmentation datasets using a set of state-of-the-art segmentation models, and compare the original definition of Panoptic Quality with the version that includes the more general matching condition. The experimental results show that although the distribution of scores for both metrics closely matches, the version of the metric with the altered matching condition yields more True Positives

than the original definition. As such, this metric could be preferable over the original definition of Panoptic Quality in cases where a high recall is required, such as in-person detection for self-driving cars, at the cost of slightly less closely matching clusters.

Chapter 7 Although the metrics discussed in Chapters 5 and 6 can be applied to a wide range of clustering tasks, our focus is on their application to text segmentation tasks, as this is a recurring task in document digitization. As with many other clustering tasks, there exists a large number of metrics for the task, with little to no consensus on the most appropriate one. To gain an insight into the advantages and disadvantages of different types of evaluation metrics, we pose the following research question:

RQ6 *What is the most appropriate type of metric for the task of text segmentation?*

To answer this research question, we evaluate three different families of evaluation metrics both empirically and theoretically, where we focus on specific properties of the evaluation metrics, such as their ability to differentiate between different predictions and their ability to assign partial credit to semi-correct predictions. Our experiments show that of the three different families of metrics evaluated, the family of metrics that compute segmentation quality based on segment-level comparisons of the ground truth and prediction behaves most consistently throughout our experiments, having the most favorable properties under different experimental conditions.

This part evaluated the quality of evaluation measures for the document segmentation and clustering tasks, and proposed novel measures better capturing the quality of document processing technology for FOIA documents.

1.2 Main Contributions

In this section we give a brief overview of the contributions made in the thesis, categorized into theoretical, empirical and artifact contributions.

Artifact Contributions

- A benchmark for the task of Page Stream Segmentation, consisting of two datasets of concatenated documents annotated with document boundaries, which we use to evaluate different approaches to the task of Page Stream Segmentation (Chapter 2).
- A dataset of Dutch decision letters to FOIA requests, with bounding box annotations of redactions, which we use to develop and evaluate three algorithms for the automatic detection of redacted text (Chapter 3).
- The *Woogle* dataset, consisting of Dutch FOIA documents acquired from over a thousand suppliers, processed using the methods developed in Chapters 2 and 3 to more closely adhere to the FAIR data principles (Chapter 4).

Empirical Contributions

- A comparison of different approaches to the task of Page Stream Segmentation in realistic conditions, where in addition to the traditional setup, model performance is evaluated on data not from the original training set (Chapter 2).
- A comparison of three different models for the automatic detection of redacted text in Dutch FOIA documents, including an analysis of the behaviour of these models when presented with documents without redactions (Chapter 3).
- A comparison of the BCubed metric with the ELM metric, both on synthetically generated data as well as on one of the Page Stream Segmentation datasets from Chapter 2 (Chapter 5).
- A comparison of the Panoptic Quality metric with our proposed metric with an altered matching criterion, on three commonly-used datasets for instance segmentation in images (Chapter 6).
- An empirical comparison of three paradigms of evaluation metrics for the task of text segmentation, with comparisons on both synthetic datasets as well as on one of the Page Stream Segmentation Datasets from Chapter 2 (Chapter 7).

Theoretical Contributions

- ELM, an adaptation of the BCubed metric that has favorable properties compared to the original metric formulation (Chapter 5).
- A more general matching condition for the Panoptic Quality metric that maintains a one-to-one mapping of predicted and gold standard items, while being more general than the original metric (Chapter 6).

1.3 Thesis Overview

In this section we provide a brief overview of the contents of the thesis and give some directions on the reading order.

The main aim of the thesis is *to develop enabling technology to improve FOIA document quality at scale, and to evaluate the quality of these extreme document clustering and segmentation tasks*. The thesis consists of eight chapters, including the introduction and conclusion chapters, and the research is divided into two parts.

Part I aims *to develop effective document processing technology for FOIA documents*, with Chapter 2 discussing suitable algorithms for the task of Page Stream Segmentation, and Chapter 3 providing a study of neural image segmentation methods for the automatic detection of redacted text in FOIA documents. Part I concludes with the creation of the Woogle dataset in Chapter 4, a large-scale collection of digitized documents and a testbed for the methods developed in the first two chapters.

Part II aims *to develop effective evaluation of extreme document segmentation and clustering tasks*. Chapter 5 details changes made to the BCubed clustering metric to address some of the shortcomings of this metric, and Chapter 6 discusses an adaptation

of the Panoptic Quality metric developed by Kirillov et al. [60] to use a more general matching condition. Part II is concluded by Chapter 7, which surveys three different families of evaluation metrics and determines their advantages and disadvantages when applied to several text segmentation tasks.

1.4 Origins

The chapters in this thesis are based on the following papers:

Chapter 2 is based on the following papers:

- R. van Heusden, J. Kamps, and M. Marx. OpenPSS: An open page stream segmentation benchmark. In *Linking Theory and Practice of Digital Libraries: 28th International Conference on Theory and Practice of Digital Libraries, TPDL 2024, Ljubljana, Slovenia, September 24–27, 2024, Proceedings, Part I*, pages 413–429. Springer, 2024. doi: 10.1007/978-3-031-72437-4_24. URL https://doi.org/10.1007/978-3-031-72437-4_24.
- R. van Heusden, J. Kamps, and M. Marx. Woor: A new open page stream segmentation dataset. In *ICTIR '22: The 2022 ACM SIGIR International Conference on the Theory of Information Retrieval, Madrid, Spain, July 11 - 12, 2022*, pages 24–33. ACM, 2022. doi: 10.1145/3539813.3545150. URL <https://doi.org/10.1145/3539813.3545150>.

The conceptualization of the paper was done by all authors of the manuscript, with the initial experiments being performed by both RH and MM. The writing of the manuscript and the conduction of the experiments and analyses presented in the paper was mostly done by RH, with both MM and JK participating significantly in discussions about the paper and the proofreading of the manuscript.

Chapter 3 is based on the following papers:

- R. van Heusden, K. Meijer, and M. Marx. Redacted text detection using neural image segmentation models. *International Journal on Document Analysis and Recognition (IJ DAR)*, 2025. URL <https://doi.org/10.1007/s10032-025-00513-1>.
- R. van Heusden, A. de Ruijter, R. Majoor, and M. Marx. Detection of redacted text in legal documents. In *International Conference on Theory and Practice of Digital Libraries*, pages 310–316. Springer, 2023. doi: 10.1007/978-3-031-43849-3_28. URL https://doi.org/10.1007/978-3-031-43849-3_28

The conceptualization of the paper was done by all authors, and the initial experiments were carried out by KM. RH continued these experiments and expanded on them by adding the results of a second neural method, and adding additional analyses on model behavior. All authors participated in the writing of the manuscript, with MM proofreading the manuscript.

Chapter 4 is based on the following paper:

- R. van Heusden, M. Larooij, J. Kamps, and M. Marx. A collection of fair dutch freedom of information act documents. *Scientific Data*, 12(1): 795, 2025. ISSN 2052-4463. doi: 10.1038/s41597-025-05052-2. URL <https://doi.org/10.1038/s41597-025-05052-2>.

All authors participated in the conceptualization of the paper, with RH, MM, JK and ML being involved in the collection of the data and the conceptualization of the paper contents. The experiments and visualizations were performed by RH and MM, and the software development was handled by ML. The writing of the original draft was performed by RH and MM, and MM and JK were involved in the proofreading and correction of the manuscript.

Chapter 5 is based on the following papers:

- R. van Heusden, J. Kamps, and M. Marx. Bcubed revisited: Elements like me. *Discover Computing*, 27(1):5, 2024. doi: 10.1007/s10791-024-09436-7. URL <https://doi.org/10.1007/s10791-024-09436-7>.
- R. van Heusden, J. Kamps, and M. Marx. Bcubed revisited: Elements like me. In *ICTIR '22: The 2022 ACM SIGIR International Conference on the Theory of Information Retrieval, Madrid, Spain, July 11 - 12, 2022*, pages 127–132. ACM, 2022. doi: 10.1145/3539813.3545121. URL <https://doi.org/10.1145/3539813.3545121>.

All authors participated in the conceptualization of the paper, with both RH and MM being involved in the conceptualization of the experiments and the methodology, with RH carrying out the experiments, and both RH and MM being involved in the formal analysis of the results. Software development was carried out by RH. Both MM and JK were involved in the proofreading and correction of the manuscript.

Chapter 6 is based on the following paper:

- R. van Heusden and M. Marx. A sharper definition of alignment for panoptic quality. *Pattern Recognition Letters*, 185:87–93, 2024. doi: 10.1016/j.patrec.2024.07.005. URL <https://doi.org/10.1016/j.patrec.2024.07.005>.

The conceptualization of the experiments was performed by both RH and MM, the original draft was made by RH, with MM involved in the corrections and proofreading of the manuscripts, as well as supporting in the project administration. The formal analysis of the results was performed by RH and MM, the investigation and experiments were performed by RH.

Chapter 7 is based on the following paper:

- R. van Heusden and M. Marx. Text segmentation metrics: A survey, 2025. To be submitted.

All authors participated in the conceptualization of the paper, with both RH and MM being involved in the conceptualization of the experiments and the methodology, with RH carrying out the experiments, and both RH and MM being involved in the formal analysis of the results. Software development was carried out by RH. MM was responsible for the proofreading and correction of the manuscript.

The writing of the thesis also benefited from work on the following publications, grouped by subject:

ParlaMint Corpus

- T. Erjavec, M. Ogrodniczuk, P. Osenova, N. Ljubesic, K. Simov, A. Pancur, M. Rudolf, M. Kopp, S. Barkarson, S. Steingrímsson, Ç. Çöltekin, J. de Does, K. Depuydt, T. Agnoloni, G. Venturi, M. C. Pérez, L. D. de Macedo, C. Navarretta, G. Luxardo, M. Coole, P. Rayson, V. Morkevicius, T. Krilavicius, R. Dargis, O. Ring, R. van Heusden, M. Marx, and D. Fiser. The parlamint corpora of parliamentary proceedings. *Language Resources and Evaluation*, 57(1):415–448, 2023. doi: 10.1007/S10579-021-09574-0. URL <https://doi.org/10.1007/s10579-021-09574-0>.
- T. Erjavec, M. Kopp, N. Ljubešić, T. Kuzman, P. Rayson, P. Osenova, M. Ogrodniczuk, Ç. Çöltekin, D. Koržinek, K. Meden, J. Skubic, P. Rupnik, T. Agnoloni, J. Aires, S. Barkarson, R. Bartolini, N. Bel, M. C. Pérez, R. Dargis, S. Diwersy, M. Gavrilidou, R. van Heusden, M. Iruskieta, N. Kahusk, A. Kryvenko, N. Ligeti-Nagy, C. Magariños, M. Mölder, C. Navarretta, K. Simov, L. M. Tungland, J. Tuominen, J. Vidler, A. I. Vladu, T. Wissik, V. Yrjänäinen, and D. Fišer. Parlamint II: advancing comparable parliamentary corpora across europe. *Language Resources and Evaluation*, pages 1–32, 2024. ISSN 1574-0218. doi: 10.1007/s10579-024-09798-w. URL <https://doi.org/10.1007/s10579-024-09798-w>.
- R. van Heusden, M. Marx, and J. Kamps. Entity linking in the parlamint corpus. In *Proceedings of the Workshop ParlaCLARIN III within the 13th Language Resources and Evaluation Conference*, pages 47–55, Marseille, France, June 2022. European Language Resources Association. URL <https://aclanthology.org/2022.parlaclarin-1.8/>.
- R. van Heusden, J. Kamps, and M. Marx. Neural coreference resolution for dutch parliamentary documents with the DutchParliament dataset. *Data*, 8(2):34, 2023. doi: 10.3390/data8020034. URL <https://doi.org/10.3390/data8020034>.

Document Processing

- F. Bakker, R. van Heusden, and M. Marx. Timeline extraction from decision letters using chatGPT. In *Proceedings of the 7th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE*

- 2024), pages 24–31, St. Julians, Malta, Mar. 2024. Association for Computational Linguistics. URL <https://aclanthology.org/2024.case-1.3>.
- L. Busch, R. van Heusden, and M. Marx. Using deep-learned vector representations for page stream segmentation by agglomerative clustering. *Algorithms*, 16(5):259, 2023. doi: 10.3390/a16050259. URL <https://doi.org/10.3390/a16050259>.
 - R. van Heusden, H. Ling, L. Nelissen, and M. Marx. Making PDFs accessible for visually impaired users (and findable for everybody else). In *International Conference on Theory and Practice of Digital Libraries*, pages 239–245. Springer, 2023. doi: 10.1007/978-3-031-43849-3_21. URL https://doi.org/10.1007/978-3-031-43849-3_21.

Part I

Document Processing Technology for FOIA Documents

2

OpenPSS: An Open Page Stream Segmentation Benchmark

Even though an increasing number of FOIA documents are now born digital, a significant number of published FOIA documents are still scanned-in copies of documents. When scanning in documents, it is common practice to scan multiple documents consecutively, to save time. The downside of this is that the output is a large PDF file without marked document boundaries, which has negative consequences for downstream applications such as a search engine, where neatly separated documents are often required.

In the literature, where this task is usually referred to as *Page Stream Segmentation* (PSS), several different models have been developed. However, a lack of publicly available training and evaluation data, in part due to the fact that much of the research is done on in-house datasets of companies, has meant that comparing these different models is rather difficult. In particular, the robustness of different models to out-of-distribution data has not been systematically evaluated. This is of particular importance to us, since the large number of different suppliers of FOIA documents means that a model should be able to handle differences in document layout, structure, language usage, and so on.

In order to create a clear overview of the efficacy of several common approaches based on Machine Learning, and to investigate their robustness in the scenario of out-of-distribution data, we pose the following research question.

RQ1 What is the efficacy of methods from Machine Learning for the task of Page Stream Segmentation?

In order to answer this question we have created two large segmentation datasets based on Dutch FOIA requests, containing both textual- and visual representations of pages, and we measure the performance of models from four different groups of models, using evaluation metrics on both the page- and document level.

This chapter was published as: R. van Heusden, J. Kamps, and M. Marx. OpenPSS: An open page stream segmentation benchmark. In *Linking Theory and Practice of Digital Libraries: 28th International Conference on Theory and Practice of Digital Libraries, TPDL 2024, Ljubljana, Slovenia, September 24–27, 2024, Proceedings, Part I*, pages 413–429. Springer, 2024. doi: 10.1007/978-3-031-72437-4_24. URL https://doi.org/10.1007/978-3-031-72437-4_24.

We evaluate the models both in an in-distribution setting, as well as the out-of-distribution setting, where a model is trained on one dataset, and tested on another dataset, in order to measure the robustness of these models when present with data from a different source.

Our results show that the group of neural models that operate under a binary classification scheme obtain superior performance, and that in the case of model performance on out-of-distribution data, a model that relies solely on the textual information contained within a document yields the best results.

2.1 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 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 [18, 46, 121]. 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 built using concatenated documents, it is possible that relevant documents are not scored properly as they might be contained within longer documents overshadowed by irrelevant content.

Although work has recently been done on applying state-of-the-art machine learning techniques such as BERT [33] and LSTM [55] models to the task of PSS [18, 46, 121], 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 chapter is organized as follows. Section 2.2 introduces the related work regarding several aspects of page stream segmentation and the current state-of-the-art. Section 2.3 discusses the construction of the dataset and the different tasks and approaches used in this chapter. Section 2.4 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 Section 2.5. We finish with a discussion and conclusion in Sections 2.6 and 2.7.

2.2 Related Work

2.2.1 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 [8, 30, 49, 78]. 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 [121] introduce a neural segmentation model based on convolutional 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 [2, 68] 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. [18] also adopted this binary classification approach, but instead focused only on the image domain. Using the EfficientNet [97] architecture instead of a VGG16 model they improve upon the scores of Wiedemann and Heyer on the Tobacco800 dataset. In a similar vein, Guha et al. [46] 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. [53], 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 [63, 77, 119], with for example Koshorek et al. [63] 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.

2.2.2 Other Datasets

Although most PSS datasets are private, two public datasets are available, namely the *Tobacco800* [127, 128] and *AI Lab Splitter* [18] datasets. The *Tobacco800* dataset consists of a single stream of 800 documents, totaling 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 chapter is an extension of previous work [104] where a single smaller dataset was used, and some preliminary experiments using only non-learned baselines were performed comparing different metrics. This chapter 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.

2.3 Method

2.3.1 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

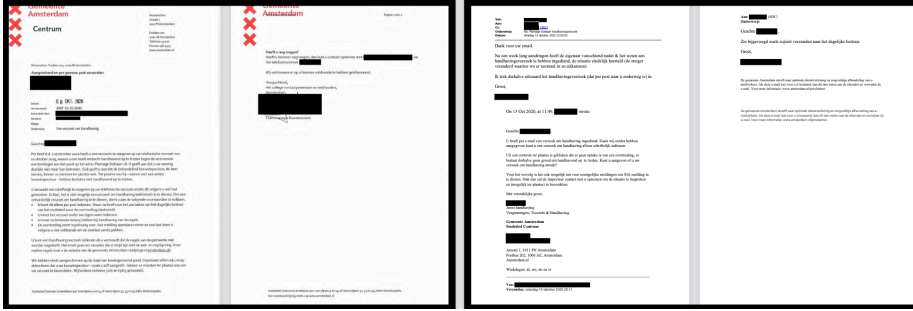


Figure 2.1: Example of a page stream from the OpenPSS-SHORT dataset with two documents, both consisting of two pages, with the black borders indicating document boundaries

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 2.1 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.

The OpenPSS-LONG dataset originated from requests to ministries during the COVID-19 pandemic, and the page streams were manually annotated with document boundaries. The OpenPSS-SHORT dataset is constructed from three Dutch governmental bodies that 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 [89] and Choi [28].

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 [91].

Table 2.1 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.

In the two existing datasets, and in particular in *Tobacco800*, single-page documents are over-represented. Because of this, so-called ‘degenerate’ segmentation algorithms [9], or algorithms that simply predict no boundaries or only boundaries can achieve deceptively high scores. Since *Tobacco800* only consists 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

¹<https://github.com/tesseract-ocr/tesseract>

2. OpenPSS: An Open Page Stream Segmentation Benchmark

Table 2.1: Overview of the key properties of the OpenPSS-LONG, OpenPSS-SHORT, Tobacco800 and A.I. LAB Splitter datasets

	Number Streams	Number Documents	Number Pages	Percentage 1 Page Documents	Median Number Pages in Stream	Median Number Documents in Stream	Image +Text
Tobacco800	1	736	1,290	.63	-	-	✓
AI LAB Splitter	1,869	5,487	31,789	.46	9	1	×
OpenPSS-LONG	110	24,181	89,491	.30	217	55	✓
OpenPSS-SHORT	312	8,162	52,177	.31	60	8	✓

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.

2.3.2 PSS Variants

In this chapter 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.

2.3.3 Models

Baselines. As fixed non-learned approaches to PSS can score remarkably well, the so-called “degenerate algorithms” [9] 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 rest of the chapter 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 [24] 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 the 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 [121], the BERT model from Guha et al. [46] and the EfficientNet from Braz et al. [18]. The TEXT-CNN model consists of a GRU model followed by a Convolutional Neural Network (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 .0001 and the model is trained for twenty epochs. The VGG16 model is pretrained on the ImageNet dataset [32], 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 .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 .00001 and a batch size of 8. To be usable for Dutch, a Dutch version of BERT was used in the experiments². 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 Labeling methods Inspired by the work of Koshorek et al. [63], 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 .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 models use either features extracted from a pretrained VGG16 model (for the image domain), or page embeddings extracted from a Dutch Doc2Vec model [66] (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 chapter are publicly available on Github³

²<https://github.com/wietsedv/bertje>

³<https://anonymous.4open.science/r/OpenPSSbenchmarkTPDL-D851/>

2.3.4 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 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 [88].

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 [64], where the output of the ensemble model is considered correct if one of the models is correct, providing an upper bound on the performance of the ensemble.

2.3.5 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 the severity of errors.

For the document-level evaluation, we report the Panoptic Quality (PQ) score, developed in computer vision [60]. 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 h , both of which are seen as sets of pages is measured by their Jaccard similarity and is called *Intersection over Union* $IoU(t, h)$. A pair (t, h) is a True Positive if $IoU(t, h) > .5$. 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 h 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

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

Table 2.2: Results of the standard PSS task for the various algorithms on the OpenPSS-LONG and 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 ($wF1$ denotes a Weighted F1)

Model	OpenPSS-LONG						OpenPSS-SHORT					
	Page			Document			Page			Document		
	P	R	F1	SQ	F1	$wF1$	P	R	F1	SQ	F1	$wF1$
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

sets consist of multiple streams, we measure the performance of models by the averages of the metrics over the streams.

2.4 Results

2.4.1 Standard Page Stream Segmentation Task

The main results of the *Standard PSS* task are shown in Table 2.2, where the models are grouped into their approach types as described in Section 2.3.3, 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 2.2 contains a lot of information, Figure 2.2 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

2. OpenPSS: An Open Page Stream Segmentation Benchmark

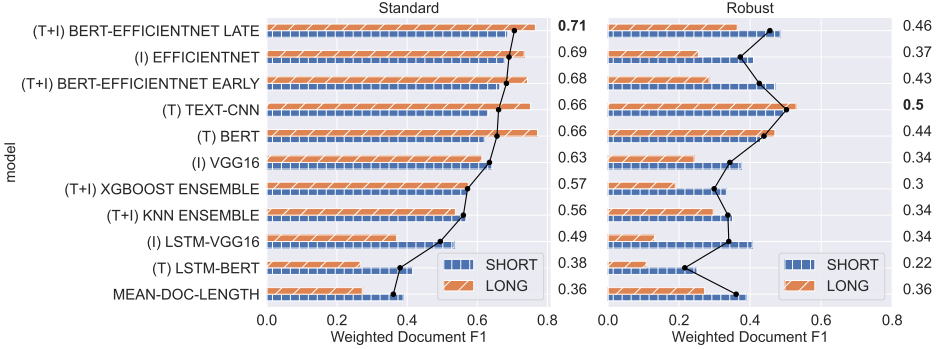


Figure 2.2: Weighted Document F1 scores for selected models on the Standard PSS task (left) and the robust PSS task (right) for both the OpenPSS-SHORT and OpenPSS-LONG datasets. Bold indicates the model with the highest average performance on both datasets. The black dots indicate the model performance averaged over both datasets.

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 uni-modal approaches for 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

Table 2.3: 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 ($wF1$ denotes Weighted F1)

Model	OpenPSS-LONG						OpenPSS-SHORT					
	Page			Document			Page			Document		
	P	R	F1	SQ	F1	$wF1$	P	R	F1	SQ	F1	$wF1$
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

enough for the task of PSS and therefore will not be included in further robustness experiments, as their baseline performance is simply too low.

2.4.2 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 2.3 shows the main results of the robustness experiments, and Figure 2.3 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 be-

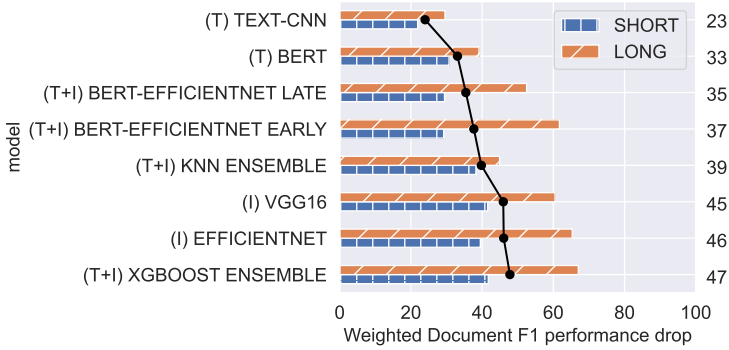


Figure 2.3: Relative performance decrease in percentages for models on the OpenPSS-LONG and OpenPSS-SHORT datasets on the robust PSS experiments where a model is trained on one dataset, and tested on the other. The performance drop is calculated as $100 \cdot (\text{standard score} - \text{robust score}) / \text{standard score}$

tween pages. However, this is less the case for the image 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.

2.4.3 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 [93], diversity is important in successful combination of models, so the correlation between the predictions of different models is also reported.

Figure 2.4 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 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

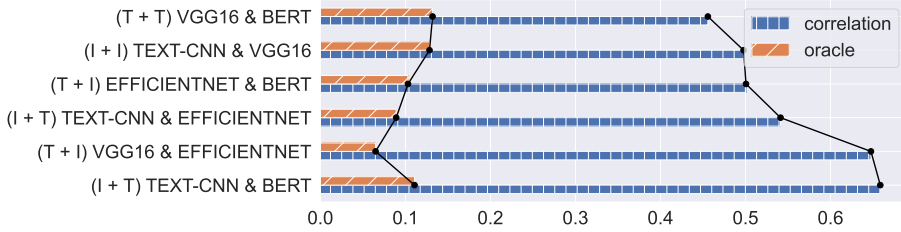


Figure 2.4: Barplot of the Pearson correlation and the difference with the oracle model in Weighted Document F1 score for six model combinations, where the (maximal) oracle score is calculated with the method described in Kuncheva [64]

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 behavior 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.

2.5 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, h) is a True Positive if the overlap between t and h is strictly larger than the non overlapping parts (formalized as $IoU(t, h) > .5$). 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 two 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 is contained within a larger predicted document, in 4% of the cases the document is dispersed over two 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 eight of every ten 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 $IoU(t, h) > .5$ requirement), and thus the overlap is perfect. For True Positives between three and ten pages long, the maximum number of non-overlapping pages can be between two and nine, 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 ten and fifty pages, the average IoU score is 0.92, and on average there is a mismatch of roughly two 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 the 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 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).

2.6 Discussion & 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 for 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 labeling 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 [114], the limitations of the method with regards to long documents can be solved, in which case its simplicity might prove useful as a practical PSS system.

2.7 Conclusion

In this chapter, we investigated the efficacy of methods from Machine Learning for the task of page stream segmentation, evaluating these models on both the standard and robust tasks using page- and document-level evaluation metrics.

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.

Redacted Text Detection Using Neural Image Segmentation Methods

Due to the nature of FOIA documents, they often contain personal information such as email addresses, names or phone numbers of public servants, or other sensitive information. Naturally, this information has to be removed from the documents before publication, a process referred to as *redaction*. This redaction can be performed using a variety of different methods, depending on the supplier, with some agencies using specialized software, simple text editor options, or even black marker pens.

Knowing how often and where these redactions occur in a document is not only useful when compiling statistics on the redaction practices of organizations, but having accurate information on the locations of these redactions can also aid in the development of text-to-speech software for these types of documents, allowing for custom behavior when a redaction is encountered.

As with the task of page stream segmentation, one of the main requirements of a system for our use case is that it can be applied to a wide variety of documents, without having to be trained separately for different data suppliers. In the case of the detection of redacted text, this not only means that the system must be able to detect a wide variety of different redaction types, but also that it must be robust to all different kinds of documents, possibly containing graphics or illustrations that look similar to redactions. With this requirement in mind, it is unlikely that a rule-based system based on textual information, as available in the literature, is capable of performing this task to a satisfactory level. Therefore, in this chapter, we turn our attention to recent developments in the Computer Vision domain, and the models developed there for the task of image segmentation and object detection, which seem like a good fit for the task of redacted text detection. As such, this chapter is centered around the following research question.

RQ2 What is the efficacy of neural image segmentation methods for the large-scale detection of redacted text?

In order to answer this research question, we created an annotated dataset of the

This chapter was published as: R. van Heusden, K. Meijer, and M. Marx. Redacted text detection using neural image segmentation models. *International Journal on Document Analysis and Recognition (IJ DAR)*, 2025. URL <https://doi.org/10.1007/s10032-025-00513-1>.

four most common types of redactions, and we compare a rule-based detection model with two current neural image segmentation methods, namely a Mask RCNN [52] and a Mask2former [27] model. To evaluate each of the methods in a realistic scenario, we additionally present the systems with a selection of pages that does not contain any redactions, and measure how often false positives occur. Our experimental results show that both the Mask RCNN and Mask2Former methods outperform the rule-based approach when it comes to correctly detecting the four different redaction types, and that, when presented with pages without redactions, the models produce only a very low amount of false positives, compared to the rule-based model.

3.1 Introduction

Text redaction refers to the process of removing sensitive information from texts which are made public. This often concerns personal information like names of witnesses in court cases, financial information from contracts, or information released through Freedom of Information Act (FOIA) requests [12, 15]. Text redaction must satisfy two balancing properties. On the one hand, it must be *safe*, text which has to be removed cannot (or only with very small probability) be recovered again from the document. On the other hand, it must be *conservative*, meaning that all other text, layout information and metadata must be kept intact.

Although this seems trivial, a large number of possible ways of redacting information exist, such as using specialized software, blurring- or mosaicing techniques, or even just a black marker pen [54]. Some redaction guidelines¹² specifically recommend printing, manually redacting and scanning a PDF document, which means all data on the document-structure and all embedded metadata is lost.

The *detection* of redactions is either a text- or image segmentation task, in which the redacted pieces of text on a page are detected and indicated by their precise bounding boxes [15]. It can be used for corpus analysis, estimating the number of redactions and the ratio between redacted- and visible text. Apart from using this information for gathering corpus statistics, it can also be used to check the quality of the redactions, and whether the text is truly redacted, or still retrievable. Several papers have been published on the vulnerability of different redaction techniques such as black bars, mosaicing and blurring, and have shown that, in specific cases, the original text can be retrieved using either a rule-based- or statistical approach [15, 54, 76] The output of a detection system can also be used when making released texts accessible according to the WCAG guidelines³. When a redacted document is read out loud by software, the redacted pieces simply become a (sometimes quite long) silence, and the spoken sentence is no longer grammatical. By using the detected bounding boxes, redactions can be given an *alt* tag, which can be used by the text-to-speech software to indicate that a piece of text has been redacted.

Text redaction detection can both be seen as a text- and as an image-segmentation

¹<https://www.nj.gov/grc/custodians/redacting/>

²<https://www.justice.gov/oip/blog/foia-post-2008-oip-guidance-segregating-and-marking-documents-release-accordance-open>

³<https://www.w3.org/TR/wcag-3.0/>

problem, partially depending on the precise type of document. Redacted texts predominantly come in the form of PDF documents. If these are “digital born”, we have access to the original text in the correct reading order. But very often, redacted texts come as PDFs consisting of *scanned documents*, and we only have access to the underlying text via Optical Character Recognition (OCR). In the latter case, an image-segmentation approach seems the first choice, whereas in the former both approaches and even their combination can be applied. For the intended applications, it is most important to correctly and precisely identify each separate piece of redacted text.

One of the few available approaches for the detection of redacted text is *Edact-Ray* [15], which approaches the task as a text-segmentation task. *Edact-Ray* is a rule-based system operating on the text of a page, together with position information of each character. Its heuristic is that when two *consecutive* words are further apart than the width of a space character, the text in between these words has most likely been redacted. The detection algorithm was manually evaluated by the authors, and had a false positive rate of 4%, with no false negatives being detected. It was reasonable to assume that the same idea could work on text obtained by OCR from scans, but, as we will show in this chapter, it does not.

van Heusden et al. [106] introduce a method for redacted text detection that combines textual information with morphological operations, and evaluate it on Dutch FOIA documents. The method sequentially removes text from an input image, and uses contour detection to detect redactions. Although the technique is successful in detecting redactions of different types, it is not robust when applied to different input sources, yielding a lot of false positives such as parts of images and logos. Although several enhancements are suggested, these would also significantly reduce the ability of the model to detect true redactions.

In an attempt to overcome the disadvantages of a rule-based approach, this chapter evaluates two neural image segmentation methods for detecting redactions. The field of image segmentation has seen a rapid development in recent years, with a multitude of models being released for a variety of tasks. These recent models can be roughly grouped into two types, those based on Convolutional Neural Networks (CNNs), and those based on Transformers [114]. Although the latter generally outperforms the former, convolutional approaches sometimes still outperform Transformer-based models on specific domains, which is why a CNN-based model is also included in this research.

The question we will answer here is: “*How well can neural image segmentation models detect redacted text?*”.

We measure performance by comparing detected redactions to ground truth redactions using the Panoptic Quality methodology [60]. We also look at the effect of the number of training examples on performance, whether we see a difference in performance for different types of redaction, and how our detectors work on “hard negatives” (pages without any redaction).

Our main contribution is a well performing (recall=.94, precision=.96) detection system based on a pre-trained Mask R-CNN model, which needs a moderate amount of training data (less than one thousand annotated pages), and which also performs well on documents without any redaction. Besides all code, we also release an extensive set of manually labeled train- and test data (1,464 pages, 11,572 redactions), grouped by redaction type (see Table 3.1). Lastly, we show that the heuristic *Edact-Ray* [15],

redaction detector does not generalize to text derived from scanned documents by OCR.

The rest of the chapter is structured as follows. In Section 3.2, we briefly discuss related work in document segmentation and recent neural instance segmentation models, as well as recent work for automatic detection of redactions. In Section 3.3, we describe the used dataset and methods in detail, including the annotation process and the specific parameters used for the neural methods. Section 3.4 contains the evaluation of the two neural models in comparison to the rule-based baseline, and also contains several examples. We conclude with a short discussion of the results and possible directions for future work in Section 3.5.

3.2 Related Work

3.2.1 Detection of Redacted Text

Although previous work on the automatic detection of redacted text is rather limited, several approaches have been proposed in the literature, which use either textual information, or a combination of both textual- and visual information to automatically detect redactions. Bland et al. [15] developed the Edact-Ray Tool Suite to better understand and fix redactions in text documents, where the first part of the pipeline detects redactions in PDF documents. Redactions are detected by identifying gaps between pairs of words that are larger than a single space character, and that consist of more non-white-than white pixels, to reduce the number of false positives. A downside of this approach is that the document should contain information on the location of characters, something that is often not present for scanned-in documents.

A technique that is more suited for usage with scanned documents is proposed by van Heusden et al. [106], who describe a method based on a combination of Optical Character Recognition (OCR) and morphological operations to perform the detection. This approach reaches an F1 of .79 with an average IoU (the segmentation quality) of the correctly identified redactions of .86 (see Table 3.3). However, the fact that the model relies on handcrafted rules means that it is difficult to adapt the model to unseen scenarios, for example when new redaction types are introduced.

3.2.2 Neural Image Segmentation

In the domain of Computer Vision, it is common to use pre-trained models such as VGG16 [95] or ResNet [51] trained on large datasets of images, and to finetune the architectures with a domain-specific dataset. This is also the case for Document Object Detection (DOD), the task of locating and identifying various types of elements in documents, for example figures, tables, paragraph heading, etc. Naturally, redacted text detection is also a DOD task.

One such model that adapts a pre-trained model for DOD tasks is Figure Caption Extract Net (FCENet), developed by Liu et al. [74]. The model architecture is based on the BlendMask architecture [22], which combines coarse object detections with fine-grained predictions based on attention. FCENet contains the addition of horizontal and vertical attention in the fine-grained detection step and the addition of separate

prediction towers for figure and caption detection to adapt it to the task of document-image segmentation. The FCENet model is compared to BlendMask, Yolact and Mask R-CNN, three other neural instance segmentation models [16, 22, 52], and it outperforms all three methods in terms of both Average Precision and F1.

Another example of a general-purpose CNN model adapted for the task of document segmentation is the model proposed by Biswas et al. [13]. The model is an adaptation of a Mask R-CNN model, adding a segmentation module to perform both object detection as well as instance segmentation. The method is compared to Faster R-CNN and Mask R-CNN models on the Historical Japanese Dataset (HJD) and PublayNet datasets. The Mask R-CNN model that is adapted for document-image segmentation outperforms both the Faster R-CNN and pre-trained Mask R-CNN models on the detection and segmentation of the majority of the selected categories, in terms of mean Average Precision (mAP).

The *DocSegTr* model from Biswas et al. [14] is a further improvement of this Mask R-CNN model, where part of the architecture is replaced with a Transformer model. The *DocSegTr* model is able to better handle the segmentation of larger image elements such as table and figures, but has slightly worse performance for small image elements.

Huang et al. [56] propose *LayoutLMV3* for various text-centric and image-centric tasks including document image classification and document layout analysis. The model is also based on the Transformer architecture, but differs from previous approaches by including textual information obtained from OCR. Both modalities are encoded using Transformers, and combined by using a multimodal Transformer. The addition of textual input means the model is better suited for tasks that require text understanding, such as the classification of individual items on receipts. The model was compared to various other models including UDoc and DiT_{BASE} [69], and outperformed these methods on the majority of the image analysis datasets used in the research.

To conclude, the task of Document Object Detection has seen a multitude of well performing models, based on the Mask R-CNN and Transformer architectures, where most models differ in the type of architecture that they use, and in which part of the model that architecture is used. As both types exhibit different behavior based on the specific dataset used, we use both a Mask R-CNN model as well as Mask2Former (a Transformer based model) in this research.

3.3 Methodology

In this section we will discuss the dataset created for this research, the annotation process, and provide brief descriptions of both the Mask R-CNN and Mask2Former models, as well as the baseline from van Heusden et al. [106]. We also briefly describe a variant of the *Edact-Ray* model, adapted for usage with scanned-in documents.

3.3.1 Data

To train and evaluate the developed models we use a dataset consisting of released FOIA requests from the Dutch government, where all documents are in the Dutch language. The statistics of the dataset are presented in Table 3.1, both in terms of the number of

3. Redacted Text Detection Using Neural Image Segmentation Methods

Van: 5.1.2e 5.1.2e <5.1.2e@minbuza.nl>
 Datum: dinsdag 14 dec. 2021 6:19 PM
 Aan: 5.1.2e 5.1.2e <5.1.2e@minbuza.nl>
 5.1.2e 5.1.2e @minbuza.nl> 5.1.2e @min
 Kopie: 5.1.2e 5.1.2e 5.1.2e
 5.1.2e 5.1.2e@minbuza.nl> 5.1.2e 5.1.2e 5.1.2e
 Onderwerp: 5.1.2e WUR - plenair debat Afghanistan

(a) Annotations with partially overlapping borders

Vragen



(b) Redactions contained in one another

Figure 3.1: Examples of the annotation of borders in the VGG Image Annotator tool. Note how yellow annotation lines tightly follow the redaction borders instead of going through any partially overlapping redactions

Redaction Type	Number of pages	Number of redactions
Black	314	3,914
Border	170	2,535
Color	83	1,660
Gray	381	3,242
No redaction	516	0
Total	1,464	11,351

Table 3.1: The number of pages and annotated redactions per redaction type in our annotated dataset

pages and the number of annotated redactions. The pages without annotations have been added to mimic a realistic scenario in which such pages frequently occur. The redactions in the dataset have been classified into four possible types, namely *Black*, *Border*, *Color* and *Gray*, with examples of each of them shown in Figure 3.2. Although not explicitly classified as separate types of annotation, the dataset also contains ‘difficult’ redactions where the lines of redactions are quite faint, or where annotations are of a particularly unusual shape, as can be seen in Figure 3.3, where a signature has been redacted. The dataset was annotated using the VGG Image Annotator tool [35] by three annotators. The redaction blocks with humanly visible gaps were annotated as separate redactions and touching blocks on separate lines were annotated as a single redaction. The lines of overlapping redaction blocks are tightly followed so that there is no overlap in annotations for those blocks, see Figure 3.1a. In cases where boxes overlap such as in Figure 3.1b, only the redaction that contained the other redactions was annotated. The annotation was thus done in such a manner that ground truth segments are never overlapping, but may touch each other. For the experiments a dataset split of 70%/30% for the train- and test set was used.

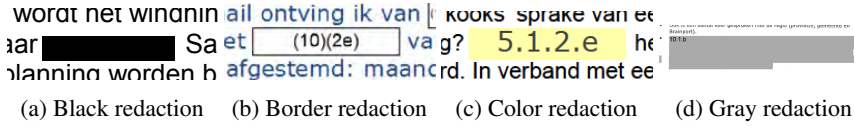


Figure 3.2: Examples of the different types of redaction in the dataset. The codes in the redactions are not type dependent. The color redaction can appear in different colors. The gray redactions can appear in different shades of gray



Figure 3.3: Redaction of a signature

3.3.2 Models

Mask R-CNN model

The Mask R-CNN model is an image segmentation model based on convolutional neural networks, developed by He et al. [52], as an extension to the Faster R-CNN model [45].

The Mask R-CNN model consists of two stages; the first stage is the Region Proposal Network (RPN) which proposes regions of interest (RoI) from input images, and the second stage extracts features from these RoIs and performs classification and bounding-box regression on them. In parallel to the second stage, the Mask R-CNN model also outputs binary segmentation masks for each RoI, allowing for both semantic- and instance segmentation. The Mask R-CNN model can be instantiated with multiple architectures; a convolutional backbone architecture for the feature extraction (over an entire image) and a network head for the bounding-box recognition and mask prediction of each RoI. For the implementation of the model we used the Detectron2 library [124] from Meta with the ResNeXt-101-32x8d [125] and Feature Pyramid Network (FPN) [72] backbones for the feature proposal and mask predictions steps respectively, following Biswas et al. [13]. We use a model trained on the ImageNet dataset [71], as this model yielded the best performance⁴. The model was trained with a learning rate of .001 without decrease, the default 1,000 warm-up iterations and a maximum of 5,000 optimization steps (roughly fifteen epochs). We did not use random flip nor did we filter out images without annotations from the training data.

Mask2Former

The Mask2Former model was introduced by Cheng et al. [27], and is an extension of the MaskFormer model [26]. The model consists of three main components, namely a backbone that extracts low-level features from input images, a pixel decoder that enhances these low-level feature maps, and a Transformer decoder module that outputs binary masks. The model can perform universal segmentation, meaning it can simultaneously perform semantic- and instance segmentation, also referred to as Panoptic segmentation. This architecture bears a resemblance to the DocSegTr model proposed by Biswas

⁴https://github.com/facebookresearch/detectron2/blob/main/MODEL_ZOO.md

et al. [14], however the generation of the feature maps is now solely performed by a Transformer model, and the generation of the final output masks combines the pixel decoder and Transformer decoder, instead of using features from the backbone. For the implementation of the Mask2Former model, we used the Mask2Former library from Meta [26] and a model pre-trained on instance segmentation for the MS COCO dataset, as this proved more successful than finetuning the pixel decoder and Transformer decoder on our dataset. We used a Swin-T model backbone [75], 5,000 optimization steps (roughly 15 epochs), a learning rate of .0001, and a batch size of two, and performed training on a single GPU.

OCR & Morphology Baseline

van Heusden et al. [106] introduce a baseline that combines OCR with morphological operations to remove all the text from an image, and subsequently performs contour detection (with some filtering on object sizes and shapes) to extract redactions from scanned-in documents. The image is first preprocessed using erosion and dilation techniques, after which OCR is used to obtain the locations of characters in the image, which are then removed. Finally, contour detection is performed to extract the remaining shapes, which are filtered based on size and orientation. We use the same hyperparameters as the original paper for our dataset as this yielded the best performance.

Edact-Ray On Scans

As previously mentioned, The Edact-Ray tool from Bland et al. [15] uses a detection method based on the location of characters and “too long” spaces in a document to detect inline redactions. This simple baseline using an appealing heuristic worked well on the digital court proceedings used in the original paper, and thus seemed a good candidate for a strong baseline. However, our redactions are in scanned documents, and so we had to adapt their method. We discuss our adaptations and argue why the heuristic does not work well on scans. For this reason, we did not include results for this model in this chapter. It is not guaranteed that character location information is present in scanned-in documents, and thus OCR has to be performed to obtain it. Moreover, some documents can contain redactions spanning multiple lines, which cannot be detected by simply using horizontal gaps between words. In an attempt to adapt the Edact-Ray detection method for scanned documents, we propose a version of the algorithm that consists of the following steps.

1. Detection of word location using OCR, using a confidence threshold of .65 to mitigate false positive text.
2. Detection of inline redaction by comparing word boundaries, and marking gaps that are larger than twice the size of an average space.
3. Detection of multiline redaction by comparing the differences in height between consecutive sentences, marking gaps that are larger than twice the average character height.

Feedback [redacted] pp 18/11 ontvangen:

(a) Inline redaction correctly detected by the Edact-Ray On Scans method



(b) Multiline redaction incorrectly detected by the Edact-Ray on scans method (redacted lines should have been marked individually)

Figure 3.4: Examples of inline- and multiline redactions identified by the Edact-Ray on scans method

4. Filtering of redactions by requiring at least half of the pixels to be non-white, to avoid false positives (for example newlines between paragraphs or text indentation).

Although the proposed approach works somewhat for the black- and color redactions (with a segmentation quality of .62 and .58 and a recognition quality (F1) of .16 and .30 for black- and color redactions respectively), the model is not able to detect border redactions, largely because of two reasons. First, the border redactions often contain codes, and as such there will not be a large gap between consecutive words (see Figure 3.1a). Although these codes could be filtered, the other problem is that these redaction boxes contain predominantly white pixels and thus will not pass through the color constraint. Lifting this constraint however is not an option, as it would lead to a large amount of false positives.

Another shortcoming of the method can be seen in Figure 3.4b. Although the model apparently correctly identified a multiline redaction, the redacted lines should have been annotated as separate redactions, not as a single block. However, since none of these lines contain any text, the model is incapable of making this distinction. This is also reflected in the scores, as the recall of the model is lower than the precision for all classes. The aforementioned problems all stem from the fact that, at its core, the algorithm relies on a sufficient amount of textual information being present to detect redactions. If however a significant amount of text has been redacted, this information is not available, and thus the model will not be able to make accurate detections, even with more sophisticated rules.

3.3.3 Mask Post-Processing

Both the Mask R-CNN and Mask2Former models output masks that can possibly overlap, which is not allowed by the assumptions underlying evaluation with the Panoptic Quality methodology. To remove these overlapping predictions, we follow the procedure described by Kirillov et al. [60], where predicted masks are sorted by their confidence and predictions with a confidence lower than a set threshold are removed. If two masks overlap more than a set threshold, the least confident mask is discarded, otherwise the overlapping portion is removed from both masks, and both masks are kept. We used a threshold of .5 for both the confidence- and overlap thresholds.

3.3.4 Computational Resources

Both the Mask R-CNN and the Mask2Former models were trained using one Nvidia Tesla P40 GPU with 24 GB DDR5 memory. The total training time for both models was roughly one and a half hours for the complete dataset, including pages without redactions.

3.3.5 Evaluation

Traditionally, Average Precision (AP) has been used to measure and compare the performance of instance segmentation models. The average precision is based on ground truth and predicted *objects* and is calculated by sorting predicted objects on their confidence score, and calculating precision and recall by starting at the most confident prediction, and including more and more samples with lower confidence. True Positives, False Positives and False Negatives are defined by using Intersection-over-Union (IoU) thresholded at .5, and average precision is then defined as the area under the precision-recall curve. Different benchmarks use slightly different definitions, by for example also taking averages over a set of IoU thresholds [71]. Regardless of the specific method, a major downside of this evaluation paradigm is that it does not measure the quality of the prediction, i.e. how close the prediction is to the ground truth. To remedy this issue, Kirillov et al. [60] have proposed the panoptic quality (PQ) metric, which, like AP, operates on ground truth and predicted objects.

The PQ metric uses the IoU to calculate matches between predicted- and ground truth objects, and defines True Positives (TP), False Positives (FP) and False Negatives (FN) over sets of objects. Given sets of ground truth and predicted objects T and H respectively, the classes TP, FP and FN can be defined as follows:

$$\begin{aligned} TP &= \{(h, t) \in H \times T \mid \text{IoU}(h, t) > .5\} \\ FP &= H \setminus \text{dom}(TP) \\ FN &= T \setminus \text{range}(TP). \end{aligned}$$

Precision, recall and F1 are then defined as usual. The F1 score is referred to in Kirillov et al. [60] as the Recognition Quality (RQ). The segmentation quality (SQ) indicates how precisely the truly predicted segments match the ground truth. It is defined as the average IoU of the set of TPs. We will report precision, recall, F1 and the segmentation quality score.

3.3.6 Code Availability

All the code and the data used in this research are publicly available on GitHub ⁵.

⁵<https://github.com/RubenvanHeusden/NeuralRedactedTextDetection>

	OCR+Morphology				Mask R-CNN				Mask2Former				
	SQ	P	R	F1	SQ	P	R	F1	SQ	P	R	F1	Support
Black	.85	.93	.90	.92	.85	.97	.98	.98	.84	.95	.95	.96	1,338
Color	.87	.84	.87	.85	.86	.97	.95	.96	.85	.97	.90	.93	542
Gray	.83	.73	.65	.69	.85	.91	.95	.93	.85	.94	.93	.93	795
Border	.88	.83	.44	.57	.86	.97	.88	.95	.83	.93	.82	.88	911

Table 3.2: SQ, Precision, Recall and F1 scores of the OCR+Morphology, Mask R-CNN and Mask2Former models on pages that contain redactions, reported per redaction type, where bold indicates the redaction type with the best score for the specific model and metric. The support is the number of redactions for each redaction type in the test set

Model	SQ	P	R	F1
OCR+Morphology	.86	.85	.72	.78
Mask R-CNN	.85	.96	.94	.95
Mask2Former	.84	.95	.90	.93

Table 3.3: Segmentation Quality, Precision, Recall and F1 of the OCR+Morphology, Mask R-CNN and Mask2Former models on pages that contain redactions (N_pages=284, N_redactions=3,586)

3.4 Results

This section contains our main results: the performance of the neural models and the baseline on the “regular” train and test set (with at least one redaction on every page) (Section 3.4.1) and on the extension of that set with hard negatives (pages without redactions) (Section 3.4.2). In Section 3.4.3 we look at the effect of reducing the number of training samples for the Mask R-CNN and Mask2Former models.

3.4.1 Redacted Text Detection

Table 3.3 reports on our main experiment: comparing the three models on the test set consisting of pages with at least one redaction. We report the segmentation quality (SQ), precision, recall and F1 scores for the rule-based OCR+Morphology, Mask R-CNN and Mask2Former models. Both neural models outperform the rule-based OCR+Morphology model on precision, recall and F1. Mask R-CNN is the best performing system concerning detection, but the OCR+Morphology model is best in precisely segmenting the redactions, witnessed by the highest SQ score.

We now zoom in and report the performance of the models for the different types of redaction exemplified in Figure 3.2. Table 3.2 shows the performance of the models per redaction type. Bold indicates the redaction type with the best score for the specific model and metric. The more traditional redaction styles, black and color, are the easiest to detect for all models. The baseline has much lower performance, in particular recall, on the gray and border types. The neural models, especially Mask R-CNN, have a

3. Redacted Text Detection Using Neural Image Segmentation Methods

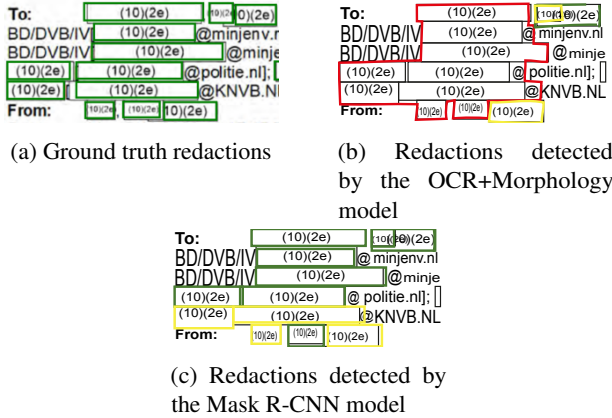


Figure 3.5: An example of border redactions being incorrectly fused by the OCR+Morphology model and correctly separated by the Mask R-CNN model. (In all figures, green indicates correct predictions, red indicates false predictions, and yellow indicates missed predictions)

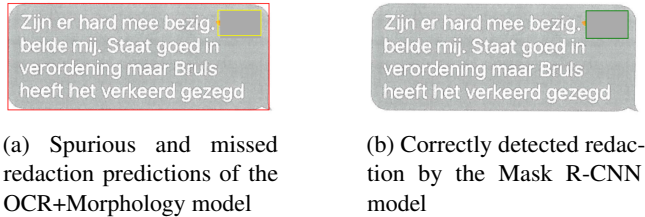


Figure 3.6: An example of a gray redaction being missed by the OCR+Morphology model, and correctly separated by the Mask R-CNN model

more consistent high performance on all four types. For the border type, the recall is remarkably lower for the OCR+Morphology and Mask2Former models.

For the border redactions, the main reason for redactions being missed was the model incorrectly fusing multiple redactions into one, as shown in Figures 3.5a and 3.5b. This type of redaction, where the redactions are very close together, occurs often for the border class, explaining the low recall of the models on this class. The Mask R-CNN and Mask2Former models handle this type of redaction much better (see Figure 3.5c), however some border redactions are still missed.

The Mask R-CNN and Mask2Former models perform more or less similar on all redaction types except for the recall on the border class which is .06 lower for the Mask2Former model. No clear cause for this difference could be found, apart from the fact that the Mask R-CNN model picked up more redactions in situations where a lot of redactions were close together. Figure 3.7 contains a challenging example where border redactions have to be recognized inside a table (with a border as well).

We did not find a singular cause for the poor performance of the OCR+Morphology

Protocol	Eigenaar
Protocol BVO's opstarten trainingen. April 2020	KNVB (10)(2e) (10)(2e) (10)(2e)
Protocol Volledig trainen betaald voetbal	KNVB (10)(2e) (10)(2e) (10)(2e)
Protocol Trainingswedstrijden	Expertgroep dit stuk KNVB (10)(2e) en KNVB (10)(2e) (10)(2e) (10)(2e) (10)(2e) (10)(2e)
Operationeel protocol veilige opstart betaald voetbal wedstrijden zonder publiek	Expertgroep dit stuk KNVB (10)(2e) en KNVB (10)(2e)* (10)(2e) (10)(2e) (10)(2e) (10)(2e)
Operationeel protocol veilige opstart betaald voetbal wedstrijden met beperkt publiek	Expertgroep

Figure 3.7: Predictions of the Mask R-CNN model on a table that contains border redactions

model on the gray redactions. There are several cases of redactions inside tables that are not detected (false negatives), but also instances in which for example gray scans of Whatsapp messages were mistaken for redactions (see Figure 3.6a).

All three models perform comparably in correctly segmenting the true positives, with SQ values between .83 and .88. (Note that SQ is always between .5 and 1 by definition of a true positive.) This is partly due to the ground truth annotations, as some of the redactions are very small, and it is difficult to place a bounding box exactly on the border of the redaction. Moreover, in the case of small objects, a discrepancy between two masks can have a large effect on the SQ metric.

3.4.2 Adding Pages Without Redactions

For this experiment, we added pages without annotations to the dataset, and evaluate the differences in model performance compared to the dataset in which every page contains at least one redaction. Of course this better resembles real-world data, which often contains pages with standard boilerplate content. We expect that the addition of redaction-free pages leads to lower recognition scores, in particular we expect more false positives. First, we added these pages without annotations to the test set only, to investigate to what extent the models were able to handle these pages without being explicitly trained with redaction-free pages. For the Mask R-CNN and Mask2Former models, this led to a significant amount of false positive detections on these empty pages (84 and 97 respectively), given that the test set contains 155 pages without redactions. As the OCR+Morphology model contains no training step and thus there is no difference with the previous experiment, it was not included in this experiment.

We now look what happens when we also train the neural models with pages without any redaction. Table 3.4 is like Table 3.3 except that now the models are trained and tested on the extension of the earlier used train- and test sets with pages without redactions. Overall performance of all three models decreases, with the decrease being largest for the OCR+Morphology method. Only the precision of the OCR+Morphology model is affected, as no additional training was performed, so the predictions on the

3. Redacted Text Detection Using Neural Image Segmentation Methods

Protocol	Eigenaar
Protocol BVO's opstarten trainingen: April 2020	KNVB bondsarts (10x2e) (10x2e)
Protocol Volledig trainen betaald voetbal	KNVB bondsarts (10x2e) (10x2e)
Protocol Trainingswedstrijden	Expertgroep dit stuk KNVB (10x2e) en KNVB (10x2e) (10x2e) (10x2e) (10x2e) (10x2e)
Operationeel protocol veilige opstart betaald voetbal wedstrijden zonder publiek	Expertgroep dit stuk KNVB (10x2e) en KNVB (10x2e) (10x2e) (10x2e) (10x2e) (10x2e)

Protocol	Eigenaar
Protocol BVO's opstarten trainingen: April 2020	KNVB bondsarts (10x2e) (10x2e)
Protocol Volledig trainen betaald voetbal	KNVB bondsarts (10x2e) (10x2e)
Protocol Trainingswedstrijden	Expertgroep dit stuk KNVB (10x2e) en KNVB (10x2e) (10x2e) (10x2e) (10x2e) (10x2e)
Operationeel protocol veilige opstart betaald voetbal wedstrijden zonder publiek	Expertgroep dit stuk KNVB (10x2e) en KNVB (10x2e) (10x2e) (10x2e) (10x2e) (10x2e)
Operationeel protocol veilige opstart betaald voetbal wedstrijden met beperkt publiek	Expertgroep

Figure 3.8: Comparison of the versions of the Mask2Former model trained only on pages with redactions (top) and on pages with and without redactions (bottom)

pages with redactions remained unchanged.

The Mask2Former model has almost no false positives on the pages without redactions, while Mask R-CNN has 52. Many of these false positives were cases where there was text inside a table, or when there was highlighted text, such as in Figure 3.9.

The number of extra false positives in the pages without redactions in the last column shows that the two neural models are more robust to realistic input with pages without redactions. The large number of false positives for the OCR+Morphology model is due to the fact that, through the text filtering and contour detection steps, it will pick up rectangular elements in pages, such as footers, parts of illustrations, and individual cells in tables.

The low amount of false positives does come with a trade-off for the Mask2Former model however, as the recall is reduced from .90 to .83. The largest decrease in recall was observed in the border class, an example of which is shown in Figure 3.8. Here we see a dramatic decrease in recall when the model is also trained on pages without redactions. The likely explanation for this is that the model has learned to ignore tables in pages (if they do not contain redactions themselves), and that this has caused the model to not detect some redactions that might be similar to tables (large border redactions with text).

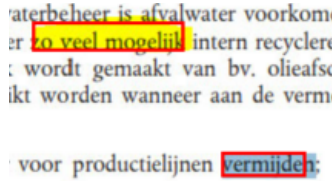


Figure 3.9: Spurious detection of highlighted text by the Mask R-CNN model (after training on the complete train set)

	SQ	P	R	F1	Empty page FPs
Rule-based	.86	.62	.70	.66	1,127
Mask R-CNN	.84	.90	.92	.91	52
Mask2Former	.84	.94	.83	.89	2

Table 3.4: SQ, precision, recall and F1 scores for the rule-based, Mask R-CNN and Mask2former models on the test set that contains pages without redactions (N_pages=439 (155 without redactions), N_redactions=3,586)

3.4.3 Influence of the Number of Training Samples

We now investigate the influence of the number of training samples on the performance of the two neural models. We take stratified subsets of the complete dataset (including pages without annotations) of 10, 20, 40, 60, 80 and 100 percent of the number of pages. We train both the Mask R-CNN and Mask2Former models on those subsets of the dataset, and evaluate on the complete test set.

Figures 3.10a and 3.10b show the progression of performance for both neural methods when the amount of training data is increased. Interestingly, the performance of both models is relatively consistent across different sizes, and even with only twenty percent of the original training set (roughly 12,200 annotations), both models achieve close to their final performance. This shows that for this specific task, the pre-training on MS COCO is very effective in training the models for instance segmentation, and that only very little data is needed to adapt these models to the specific task of redaction detection.

3.4.4 Post-processing Model Output

Although both the Mask R-CNN and Mask2Former models show a significant improvement over the rule-based model when applied to pages that contain no redactions, they still make some mistakes. Some (like the one shown in Figure 3.9), could be relatively easily removed by performing OCR on the model output, and filtering the predictions based on this. Since the Mask2Former model only had two false positives, we will focus on the Mask R-CNN model for this experiment. Of the 52 false positives yielded by the Mask R-CNN model, 22 of these false positives contained readable text according to Tesseract (with a confidence threshold of .70), and could thus benefit from

3. Redacted Text Detection Using Neural Image Segmentation Methods

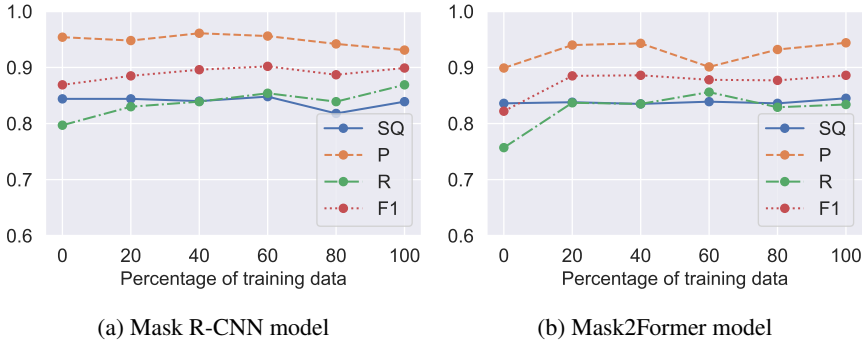


Figure 3.10: Performance of the Mask R-CNN and Mask2Former models in terms of SQ, P, R and F1 for 10, 20, 40, 60, 80 and 100 percent of the training data (in terms of number of pages)

a post-processing approach. Because some of the legitimate redactions in the dataset contain text (such as a code, or the reason for redaction) we cannot simply filter out all redactions that contain text, but rather have to use regular expressions and (fuzzy) string matching to filter out often-used codes and phrases. Using this approach, 21 of the 22 false positives detected by the Mask R-CNN model could be filtered out. However, this does come at the cost of having more false negatives, with 585 false negatives for the Mask R-CNN model after filtering, compared to 468 false positives in the original setting. A substantial portion of these false negatives came from redactions where Tesseract was not able to extract text due to the small size of the redaction (such as the bottom-left redaction in Figure 3.5c.) Although the precision of the model is increased from .90 to .96, the recall of the model has dropped from .92 to .84 resulting in a decrease of the F1 score of the model from .91 to .89. Although the drop in performance is relatively small, and in cases where precision is important this approach might be reasonable, it introduces another layer of complexity in the detection pipeline, where specific rules have to be crafted to strike a balance between false positives and false negatives.

3.5 Discussion & Future Work

Although both neural methods outperform the OCR+Morphology model in both experiments, they only operate on the images of pages, and do not explicitly use the textual information. We tried incorporating textual information using Tesseract as a post-processing step to improve model performance, but found that this approach resulted in a net decrease in model performance.

Future work in this direction could look into incorporating textual information in a more sophisticated manner, for example by using a multi-modal approach where a textual- and visual model are trained simultaneously to recognize when text in a redaction is acceptable (being a code for example), or when this indicates a false positive.

3.6 Conclusion

In this chapter, we set out to investigate the efficacy of modern image segmentation models for the task of redacted text detection, measured on a collection of annotated Dutch FOIA documents. We compared two neural instance segmentation models with a strong rule-based baseline from the literature. Models were trained and tested on documents released after a request based on the Freedom of Information Act. Both neural methods significantly outperform the rule-based method: they pick up more redactions, make fewer mistakes and are also more robust to realistic data containing pages without redactions. The Mask R-CNN model performed best, with a precision and recall of .96 and .94 respectively when trained and tested on pages with redactions. This dropped slightly to .90 and .92 when adding hard negatives in the form of pages without redactions. We additionally conducted an experiment to filter the output of the Mask R-CNN model using Tesseract to reduce the number of false positives. We found that although effective in reducing false positives, this approach also increased the number of false negatives, resulting in a net drop in performance of the resulting model. We ported the simple and appealing rule-based baseline from Bland et al. [15], which worked well on digital documents, to scanned documents, but found that the used heuristic is too brittle for this more “dirty data,” leading to subpar performance compared to the other methods.

4

A Collection of FAIR Dutch Freedom of Information Act Documents

As we alluded to in the introduction, the collection and publication of FOIA documents is an involved process, especially when done on a large scale with many different data suppliers, and where one has to deal with different organizations all having slightly different formats and standards. Using the techniques discussed in the previous two chapters, as well as others, we set out to create a large collection of Dutch FOIA documents, which we refer to as the *Woogle* dataset. This dataset is enhanced so that it has a consistent set of metadata, as well as data that adheres to FAIR data principles. By creating this dataset, we hoped to learn more about the process of digitizing such a collection, the practical challenges, and the opportunities such a dataset can bring.

Therefore, we aim to answer the following research question in this chapter.

RQ3 What lessons can be learned from a Living Lab of FOIA documents?

We answer this question by collecting FOIA documents from over a thousand suppliers using automatic web-scraping, and by using the techniques we explored in Chapters 2 and 3, as well as others, to standardize the contents and the metadata of the collection. Through the development and the usage of the collection in several research projects, we have been able to learn some valuable lessons from this Living Lab, most notably that, with the right tools and protocols, it is possible to create a large high-quality document collection, without the need for expensive specialized software or infrastructure, and that this resulting dataset is a valuable resource for research into these FOIA documents.

4.1 Background & Summary

Like over a hundred countries worldwide, the Netherlands has Freedom of Information Act (FOIA) legislation, requiring governmental institutions to release documents related to their decision-making process to the public, either passively or proactively [99]. Generally speaking, the aim of this kind of legislation is to increase government transparency, allowing citizens to examine the decision-making process, addressing possible

This chapter was published as R. van Heusden, M. Larooij, J. Kamps, and M. Marx. A collection of fair dutch freedom of information act documents. *Scientific Data*, 12(1):795, 2025. ISSN 2052-4463. doi: 10.1038/s41597-025-05052-2. URL <https://doi.org/10.1038/s41597-025-05052-2>.

inconsistencies, and therefore facilitating the proper functioning of the democratic process [94, 123].

Apart from the benefits for the democratic process, a collection of these FOIA documents can also be valuable for different disciplines within the research community. However, the current creation- and publication process of these kinds of documents in the Netherlands makes it very difficult to use this potentially valuable resource in research, as it often fails to adhere to the FAIR data guidelines outlined by Wilkinson et al. [122]. Currently, each agency is responsible for publishing their own records, usually on their own platform or website, with little to no coordination between agencies on what kinds of metadata are included, and no standards on text quality and machine-readability. The result of this is that conducting large-scale research across different agencies is complicated, as the data needs to be collected from multiple sources, metadata needs to be standardized, and there is no guarantee that the quality of the data is sufficient for usage in research.

In an attempt to mitigate the aforementioned problems, and to make this valuable resource of FOIA documents available as FAIR data, we created the publicly accessible *Woogle* dataset. The Dutch FOIA is abbreviated as *Woo* and we ensure that *Woo*-data is easily searchable, hence the name.

By creating a uniform set of metadata in a standardized format (e.g., ISO-dates for all date related events, a fixed set of document types), we facilitate both the *findability* and the *interoperability* of the document collection. The *accessibility* of the documents is handled by not only having the documents be freely available, but also by having persistent metadata, even if the source of the document no longer exists. Finally, the *re-useability* of the dataset is addressed by having high-quality machine-readable text available for the documents in the collection.

In 2022, the Dutch FOIA legislation was broadened significantly, to include not only *passive* requests of documents (a request sent to a specific government agency), but also requiring the *active* release of documents. This means that governing bodies are obliged to release these documents on a public website satisfying quite strict technical criteria, which resemble the FAIR principles.

In this chapter, we limit ourselves to the portion of the *Woogle* dataset containing passive requests, as the collection of documents resulting from active release is currently far from complete, and internationally rather unique. However, the process of collection and FAIRification of these documents is identical to that of the passively released documents.

Having such a broad collection of documents from a large number of organizations provides a wealth of opportunities for multiple purposes. For instance, searching for protected animal species like wolf, certain toads or deer yields hundreds of hits and shows when, where and in what context governing bodies deal with this topic. Besides facilitating search across different agencies, the collection also provides a wealth of information for researchers in both the computer science and political science domains, allowing for example the large-scale study of government policies on public housing or refugees, or studies on the popularity of certain keywords or topics in the requests for information. For the field of Natural Language Processing (NLP), it not only opens up the possibility of several forms of text analysis, but also allows for the training of new language models that are better tailored to this specific domain. This can in

turn be used to provide improved interaction with these documents, with for example simplification or automatic summarization of the information, making this type of information accessible to a larger number of people (which after all is their main purpose).

Although the source language of the Woogle dataset is Dutch, the usefulness of the dataset is not limited to Dutch research efforts only, and the dataset can also be of value to the international research community. The presence of high-quality metadata means that it is possible to filter documents based on specific criteria, such as a certain time period, or a certain organization, and then further process those files. One can for example select only files from the Dutch parliament during the COVID-19 pandemic, and then use automatic translation to translate the documents into another language for further analysis. The advantage of this corpus is thus not only the information itself, but also its structured format, which avoids having to manually retrieve this data, which can be especially troublesome for someone not familiar with the structure of Dutch governing bodies, as well as avoiding having to translate the entire corpus, where now only translating a subset would be sufficient.

Throughout Europe and the United States, there are several similar initiatives and collections regarding released FOIA documents, both on the national- and regional levels. Examples of regional collections and initiatives can be found in Hamburg, where a search engine for government documents was launched in 2014, and Brussels, where a similar effort to digitize FOIA documents was undertaken.^{1 2} An example of a national database of FOIA documents is the Norwegian Electronic Public Records (OEP) platform, which contains roughly 70 million documents from municipalities, ministries, and several other governmental agencies.³ The platform allows users to search the collection and maintains an agenda of political meetings for which documents are available in the repository. In a similar vein, the European Parliament has also created an online portal for accessing previously released documents, and provides means for citizens to request information that is not (yet) available online.⁴

There are also several related initiatives regarding the publication of FOIA documents in the United States, the two most prominent examples being the FOIArchive from the History-Lab project and the collection of documents released as part of the FOIA Project from the TRAC research center at Syracuse University.^{5 6} The History-Lab collection consists of eight subcorpora, totaling roughly three million documents ranging from 1861 to 2013, and is among the largest collections of de-classified government documents in the United States. Apart from the ability to search the document collection, the corpus also contains automatically extracted topics and Named Entity annotations. The FOIA Project dataset from the TRAC research center consists of roughly 9,000 documents relating to decisions made on the withholding or releasing of data by the United States federal government, including formal decisions as well as the results of lawsuits regarding publication. Apart from merely having these FOIA

¹<https://transparenz.hamburg.de/ueber-un>

²<https://transparencia.be>

³<https://einnsyn.no/statistikk/generell>

⁴<https://www.europarl.europa.eu/RegistreWeb/home/welcome.htm?>

⁵<http://history-lab.org>

⁶<https://foiaproject.org>

documents available as online information, the dissemination of the data also facilitates the development of more complex analyses of different aspects of the data. An example of this is the COVID-19 Archive Prototype, which is a collection of emails from an American medical expert, obtained through FOIA requests, and extracted from the History-Lab collection.⁷ The data was processed so that it can be filtered not only by content, but also by automatically extracted topics. A part of the History-Lab collection was also used for the development and testing of a system for automatically extracting important events in government documents using Machine Learning techniques [20].

The rest of the chapter is organized as follows. Section 4.2 starts with an explanation of the used model of FOIA dossiers, and its operationalization used in the dataset. We continue with an overview of the dataset, the collection process, and the FAIRification process. Section 4.3 contains a detailed description of the different types of information included in the dataset, and Section 4.4 elaborates on the validation performed on the quality of the dataset. We conclude with some final notes on best practices for using the dataset in Section 4.5 and the usage of third-party software in Section 4.6.

4.2 Methods

4.2.1 Dataset Structure

In the dataset, we follow the structure of a typical FOIA request, where each received request is considered a *dossier*, containing two mandatory files (the original request and a decision letter, detailing whether the request was granted or denied, and for what reasons) and, depending on whether or not the request is granted, an inventory list of released documents, and the released documents themselves (often partially redacted).

Both the dossier and the documents contained within it come with a standard set of metadata. A dossier typically has the following metadata associated with it. First, the date of the original request and the date of the decision letter. Second, the actual request (information need) and the decision (granted or denied). Key metadata of documents are whether they are fully or partially made public (that is, whether they contain redacted text), the legal reasons applied to refuse publication of certain parts of the text, and the kind of document released.

The advantage of this document structure is that it is somewhat universal, at least from the perspective of having a dossier structure associated with a request. The same model has also been applied to a collection of Estonian FOIA documents, and it resembles the model of a structured literature search, with distinct groupings of documents into clusters [17, 116].

4.2.2 Dataset Overview

The dataset presented in this chapter is a static version of the documents in the Woogie search engine on 04-11-2024. The portion of the Woogie dataset relating to the passively released FOIA documents contains 13,529 dossiers, 121,175 documents and

⁷<https://covid19-prototype.history-lab.org>

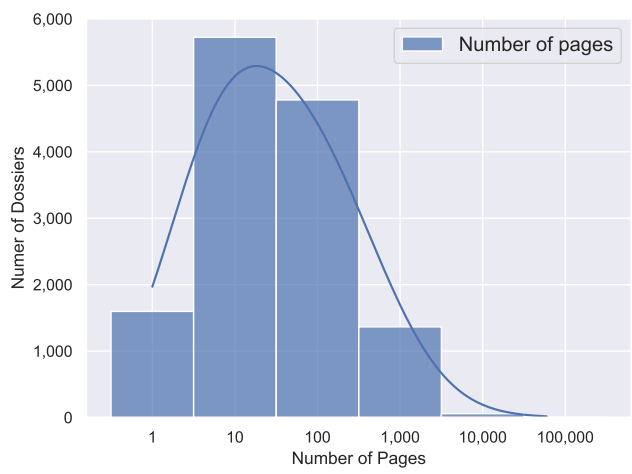


Figure 4.1: Distribution of the number of pages per dossier, with log-x scale (N=13,529, $\mu = 158$)

Document Type	Number of Pages	Number of Words
Released Documents	1,825,461	322 million
Decision Letter	245,349	47 million
Original Request	25,554	4 million
Inventory List	25,736	4.5 million
Total	2,122,100	377.5 million

Table 4.1: The number of pages and number of words for the four different document types in the dataset.

2, 122, 100 pages. The dataset contains around 378 million words, with 4.2 million unique words. The dataset contains documents released between 2001 and 2024.

Figure 4.1 gives an overview of the distribution of the number of pages per dossier, and Table 4.1 gives an overview of the distribution of pages and words over the four documents types discussed above.

Figure 4.2 provides an overview of the types of suppliers of dossiers, as well as the number of released dossiers through time. The majority of the dossiers originate from either ministries or municipalities, and the majority of dossiers have been released over the last five years. The inner ring shows that the production of dossiers within types of governing bodies is rather balanced and not Pareto distributed.

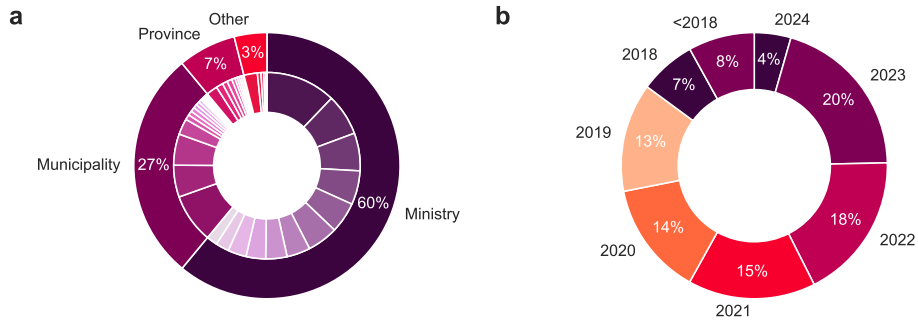


Figure 4.2: Distribution of dossiers of 59 suppliers (inner ring), grouped by the type of organization (outer ring) (a), Distribution of the number of dossiers released each year (from 2001 to 2024) (b)

4.2.3 Dataset Collection

All the FOIA dossiers released by Dutch ministries are published on a central open government platform.⁸ On this platform, filters can be applied to download only the passively published documents (*Beslissing op Wob-/Woo-verzoeken* in Dutch), and the results can be filtered to include only documents from ministries. Several municipalities are subscribed to the *OpenPub* platform, where they publish their FOIA documents, which can be accessed through an API.

Most governing bodies which are not local governments use somewhat standardized centrally provided publication software, where the information can be extracted in a way similar to the way in which the ministry documents are extracted. For other government bodies, custom scraping software was built to extract the information from the respective agency website.

A more detailed description of the scrapers and their source code is also available (in Dutch) on GitHub as well as in the *DatasetCollection* notebook present in the dataset repository.⁹ According to Dutch law, documents released under the Freedom of Information Act are by default subject to the CC-BY license, allowing re-use, if appropriate credit is given.

4.2.4 FAIRification

We strive to collect all basic metadata described above in the FOIA dossier model in our scraping process. If it is consistently made available, we collect it and normalize the data (converting different dating conventions to ISO format, classifying different requester types). The type of the document (decision, request, released document and inventory list) is not always explicitly listed, and thus has to be extracted using heuristics based on the filename and file-type. We attempt to obtain machine-readable text from the documents using the *pdftotext* software if a text layer is available in the

⁸<https://open.overheid.nl/>

⁹<https://github.com/wooverheid/WoogLeDocumentatie>

PDF document.¹⁰ If this is not the case, or if the quality of the original text is too poor, Optical Character Recognition (OCR) software is used to extract text from the images. Tesseract version 5 with Sauvola binarization is used for OCR.¹¹ Detecting whether a document needs to be OCRed is also a heuristic process performed with the *OCRmyPDF* software.¹² We store both the original text (obtained from the PDF using *pdftotext*), and the text obtained by Tesseract.

Text Quality

For assessing the overall text quality of a document, it is meaningful to have a single score indicating the machine-readability of that document. We refer to this score as the *FAIRIScore*, which rates the overall readability of a document according to five classes from A to E, where E is the lowest and A is the highest. The explanation of the classes is given below. The name, the 5 FAIRIScore values, the colors and the assignment of values to items have been inspired by the Nutri-Score [21]. To determine whether a page is scanned in, we use the *MuPDF* software to determine whether or not a page contains an image that covers (almost all of) the page, indicating a scan.¹³ Note that there is no standard procedure to test whether a PDF document is “digital born”, meaning that it has been created for instance by an export-as-PDF command in Word. Therefore we must resort to these heuristic methods.

- **A** None of the pages in the document has a page-covering image, and the original text of the document contains at least 100 characters on every page.
- **B** The document contains some (but not only) page-covering images, but the average Jaccard similarity between the original text and the OCRed text over all pages in the document is larger than .9. Also true if the document contains no page-covering images, but some pages have less than 100 characters in the original text.
- **C** The document contains some (but not only) page-covering images, and the Jaccard similarity between the original text and the OCRed text over all pages of the document is less than .9 but higher than the mean similarity over all documents.
- **D** The document contains some (but not only) page-covering images, and the Jaccard similarity between the original text and the OCRed text over all pages of the document is less than the mean similarity over all documents.
- **E** The document contains no machine readable letters in the original text, and all pages in the document consist of page-covering images.

Figure 4.3 gives an overview of the FAIRIScore distribution for all the documents released by ministries. In general, the quality of the documents released by the ministries

¹⁰<https://www.xpdfreader.com/pdftotext-man.html>

¹¹<https://tesseract-ocr.github.io>

¹²<https://ocrmypdf.readthedocs.io/en/latest/>

¹³<https://mupdf.com>

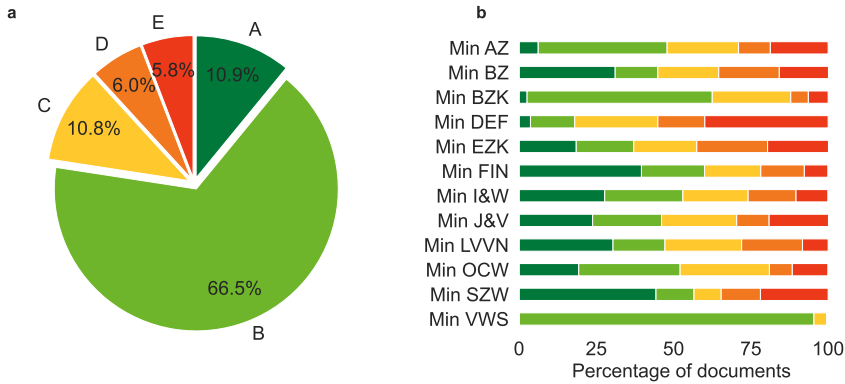


Figure 4.3: Distribution of the FAIRScore for all FOIA documents released by ministries (a) Distribution of the FAIRScore for all FOIA documents released by ministries, grouped by ministry (b)

is reasonable, with more than 75 percent of the released documents having a score of either A or B. There is quite some variability between the different ministries however, as shown in Figure 4.3, where for some ministries the number of documents of lesser quality (score C or below) being more than half of the total number of documents. We note that we deliberately have chosen not to include any automatically testable PDF-accessibility criteria as described in PDF/UA and WCAG 2.1 because the vast majority of the Dutch FOIA PDF documents does not adhere to these.

Page Stream Segmentation (PSS)

During the construction of the dataset, we found that in more than 90 percent of the cases where documents were being scanned in, multiple documents were scanned-in consecutively, and ended up being saved as one large PDF file. Naturally this is undesirable, as we want the individual documents to be findable, and not only have them as part of a very large PDF file, without document-specific metadata. The task of reconstructing the original documents from these *streams* of pages is known in the literature as Page Stream Segmentation (PSS), and several approaches based on Machine Learning techniques have been proposed for this task. Most recent approaches to this task use methods based on neural networks to classify individual pages on whether or not they start a new document, using both textual- and visual features [18, 46, 121]. The segmentation of the documents in the Woogole dataset is performed using the text-based method from Wiedemann and Heyer [121]. The reason behind using this approach is that previous research has shown that this leads to models that are more robust to input from different sources (as opposed to image-based models), which is important for the Woogole dataset, as there is a large number of different document suppliers [110].

Figure 4.4 shows the distribution of the number of documents of the *Released Documents* class in a dossier, after the application of the segmentation algorithm. We focused on this class as it is most prone to containing document scans, given that the documents might come from various different sources, whereas decision letters are

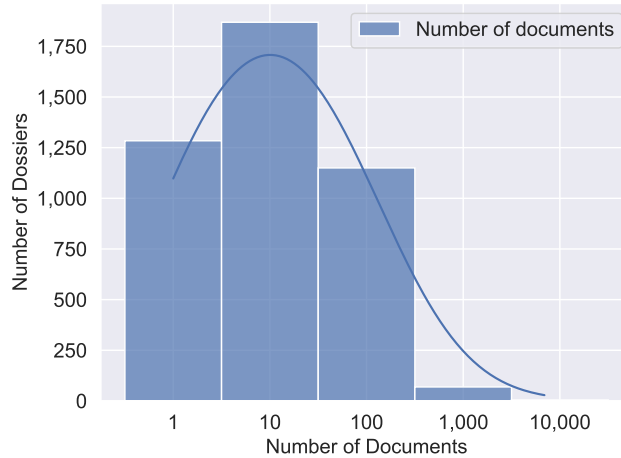


Figure 4.4: Distribution of the number of documents of the *Released Documents* class per dossier after applying PSS ($N=4,377$, $\mu = 44$, log-x scale).

often a single document created via text-processing software. After applying PSS to the *Released Documents*, the number of documents increased from 37,570 to 192,663. The median document length decreased from 26 to 5 pages, and the longest document is now 2,755 pages long, compared to 4,401 pages before segmentation. The algorithm was evaluated in previous work, where it achieved an F1 score of .78 measured on two datasets [110].

Redacted Text Detection

As some of the information in the documents is confidential or privacy-sensitive, organizations will redact the confidential information from the document before releasing it to the public. However, as there is no clear standard for redaction practices or redaction software, a plethora of different methods is used, from automatic redaction using regular expressions to manual redaction by pen. Although some of these methods retain the original metadata of the document (such as the embedded text), other methods, such as the printing, redacting and scanning techniques, do not preserve this information. As a result, text-to-speech software might not know how to properly deal with these redactions, and thus a PDF containing text redaction might be unintelligible when read aloud by a computer. To provide an overview of the extent of redaction in the dataset, we measure the amount of redacted text using Machine Learning techniques. The used algorithm for detecting redacted text is described in van Heusden et al. [106].

For the 1,172,713 pages for which the analysis was performed, 523,603 of them (around 45 percent) contained at least a single redaction, with a total of 4,359,857 redactions on these pages (around eight redactions per page). On pages that contained at least a single redaction, the median percentage of redacted characters was around eighteen, and around two percent of these pages were completely redacted. Figure 4.5 shows examples of the different styles of redactions in our corpus. Figure 4.6 shows the

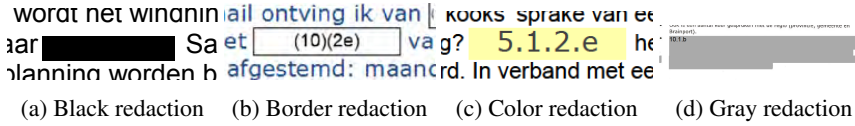


Figure 4.5: Examples of the different types of redaction in the dataset. The codes in the redactions are not type-dependent. The color redaction can appear in different colors. The gray redactions can appear in different shades of gray

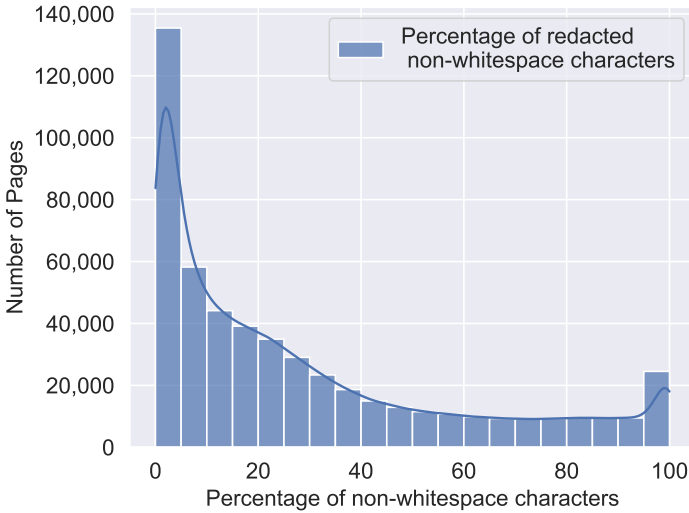


Figure 4.6: Distribution of the percentage of non-whitespace characters on a page that are, for pages that contain at least one redaction (N=523,603, median=18%)

distribution of the portion of the characters on a page that is redacted. The performance of the algorithm is evaluated in previous work, with an F1 score of .77 on a manually annotated test collection [106].

4.3 Data Records

The dataset is available at the DANS Data Station for Social Sciences and Humanities [111]. The data presented in this chapter is stored in three separate files in the *dataset* folder, namely the *woo_dossiers.csv.gz*, *woo_documents.csv.gz*, and *woo_bodytext.csv.gz* files, containing the dossier, document, and page metadata, respectively. Apart from this, the actively released documents are also stored in the dataset as extra material in the *EXTRA-active-release* folder. The dataset is available under the CC BY 4.0 license, allowing users to use and modify the data for their own purposes (giving appropriate credit).¹⁴

¹⁴<https://creativecommons.org/licenses/by/4.0/deed.en>

The dossiers, documents and pages all have separate sets of metadata associated with their records, which are stored in a separate file for each of these types. The dataframes containing the records all adhere to the Boyce-Codd Normal Form (BCNF) for databases. In these dataframes, the *dc_identifier* provides a unique ID for each of the rows in the dataframe, and for the document and bodytext dataframes the IDs of the dossier and documents respectively are included to allow the joining of the different dataframes. The precise contents of the dataframes is detailed in Tables 4.2, 4.3 and 4.4. The majority of these attributes are extracted automatically during the scraping process, and some attributes are added after processing of the document, as those discussed in the previous section. In the tables we only report on the missing values if they occur in at least five percent of values for that attribute.

In addition to the datafields mentioned in Table 4.3, the dataframe with document metadata contains an additional 9 metadata fields, all relating to metadata extracted from PDF files using the *pdfinfo* tool.¹⁵ As these fields have data values for a very small portion of the documents (between one and two percent of all documents), and will generally not be available for scanned documents, we did not include them in the description of the dataframe. As previously mentioned, the *foi_bodyText* attribute contains the text of a page (if this was present, otherwise the text is extracted using OCR), together with layout information of the text. For most use cases, such as constructing a search engine, this information will be sufficient, however there might be instances in which the original PDF has to be accessed. The *dc_source* attribute contains the URL from which the document was originally retrieved. In some cases, this URL has become invalid, for example because the structure of a website was changed. In this case the *dc_identifier* attribute can be used to retrieve the original document from the the project server using `https://doi.wooverheid.nl/?doi=` combined with the *dc_identifier* of that particular record, for example `https://doi.wooverheid.nl/?doi=n1.gm0148.2k.2020.396.doc.1` (a notebook with more information about this process is present in the dataset).

4.4 Technical Validation

When evaluating the quality of a dataset, there are several key dimensions that should be considered. Two dimensions that are commonly considered are *Consistency/Validity* and *Completeness* [83, 85]. Here consistency or validity refers to whether the values in a dataset are valid for that type of data (right datatype, in the right range), and completeness refers to how much of all data is captured in the dataset.

We further decompose the completeness of our dataset into *extrinsic* and *intrinsic* completeness, where extrinsic completeness refers to how much of the dossiers that exist are present in the dataset, and intrinsic completeness refers to the amount of missing values in the dataset.

In our case, the extrinsic completeness of the dataset is hard to measure, as the variety of different suppliers and websites makes it difficult to judge how much of the suppliers we have dossiers for. For the municipalities we have dossiers of 31 municipalities, out of all 342 municipalities in the Netherlands. For the ministries we have dossiers for all

¹⁵<https://www.xpdfreader.com/pdfinfo-man.html>

4. A Collection of FAIR Dutch Freedom of Information Act Documents

Field	type	Description	% NaNs
<i>dc_date_year</i>	date	Year when the dossier was released	-
<i>dc_description</i>	string	Short description of the dossier contents	.18
<i>dc_identifier</i>	string	Unique dossier identifier	-
<i>dc_publisher</i>	string	Code of the dossier publisher	-
<i>dc_publisher_name</i>	string	Full name of the publisher	-
<i>dc_source</i>	string	URL pointing to the source of the dossier	.07
<i>dc_title</i>	string	Title of the dossier	-
<i>dc_type</i>	string	Type code of dossier	-
<i>dc_type_description</i>	string	Human-readable version of the <i>dc_type</i>	-
<i>foi_decisionDate</i>	date	Date of decision on request	.17
<i>foi_decisionText</i>	string	Explanation of decision	.98
<i>foi_isAdjourned</i>	string	Whether or not the request is adjourned	.98
<i>foi_publishedDate</i>	date	Date of the publication of the dossier	.17
<i>foi_requestDate</i>	date	Date of original request	.88
<i>foi_requester</i>	string	requester type (organization, individual)	.99
<i>foi_retrievedDate</i>	date	Date on which the dossier was retrieved	-
<i>foi_valuation</i>	string	Decision on a request	.98

Table 4.2: Overview of the fields in the *woo_dossiers.csv.gz* archive, where every row describes a dossier. The *dc_identifier* attribute is the primary key

Field	Type	Description	% NaNs
<i>dc_format</i>	string	Format of the document (MIME type)	.38
<i>dc_identifier</i>	string	Unique document identifier	-
<i>dc_source</i>	string	Source of the document (document URL)	.40
<i>dc_title</i>	string	Title of the document	.49
<i>dc_type</i>	string	Type of document	-
<i>foi_dossierId</i>	string	Identifier of the corresponding dossier	-
<i>foi_fairiscoreVersions</i>	string	FAIRIscore of the document	-
<i>foi_fileName</i>	string	Filename of the document	-
<i>foi_nrPages</i>	int	Number of pages in the document	-

Table 4.3: Overview of the fields in the *woo_documents.csv.gz* archive, where every row describes a document. The *dc_identifier* attribute is the primary key, and the *foi_dossierId* is the foreign key

of them, as these are published on a single platform, and can be scraped with a single script. Our main focus is to ensure that, for the publishers that we have scrapers for, we collect all of the dossiers that are published on their websites. We do this by, apart from having scrapers to update the dataset daily, periodically re-scraping the websites to catch any publications that might have had incorrect dates.

For all the dossiers that we collect, we attempt to retrieve the metadata listed in the tables in Section 4.3. If this data is available, we parse the data, converting it into the

Field	Type	Description	% NaNs
<i>foi_bodyText</i>	string	Text of the page extracted using <i>pdftotext</i>	.23
<i>foi_bodyTextOCR</i>	string	Text of the page extracted using Tesseract	-
<i>foi_bodyTextJaccard</i>	float	Jaccard similarity between <i>foi_bodyText</i> and <i>foi_bodyTextOCR</i>	-
<i>foi_charArea</i>	int	Number of pixels on the page belonging to a character	.45
<i>foi_contourArea</i>	int	Number of pixels on the page belonging to a redaction	.45
<i>foi_documentId</i>	string	Identifier of the document to which the page belongs	-
<i>foi_hasOCR</i>	bool	Whether there is OCR text available	-
<i>foi_imageArea</i>	float	Percentage of a page that is covered by an image	-
<i>foi_imageCoversFullPage</i>	bool	Whether a single image completely covers an image	-
<i>foi_isFirstPageOfNewDoc</i>	bool	Whether a page is the start of a new document (PSS analysis)	.43
<i>foi_nrRedactedRegions</i>	int	Number of individual redacted regions	.45
<i>foi_pageNumber</i>	int	The number of the page within a document	-
<i>foi_percentageCharAreaRedacted</i>	float	Percentage of the characters on a page that is redacted	.45
<i>foi_percentageTextAreaRedacted</i>	float	Percentage of the total text area that is redacted (including whitespace)	.45
<i>foi_redacted</i>	bool	Whether there is redaction present	.45
<i>foi_textArea</i>	int	Number of pixels on a page which belong to text	.45

Table 4.4: Overview of the fields in the *woo_bodytext.csv.gz* archive, where every row describes a page. The *foi_documentId* is the foreign key, and the *foi_documentId* combined with the *foi_pageNumber* attribute is the (composite) primary key

required format. This includes converting dates to ISO format, standardizing dossier- and document types into the pre-defined sets of allowed values (e.g., a document type must belong to the four discussed types) or translating fields to binary attributes. To ensure the validity of the data, this parsing includes validation of the input, i.e. ensuring that parsed dates and numerical values are within the expected range, and that binary attributes only contain boolean values.

Reporting on the intrinsic completeness of the dataset, we briefly comment on all fields with at least five percent missing values. We start with the missing values in Table 4.2. The *dc_source* attribute points to the original source (URL) of the document, and in cases where PDF files were submitted directly to us by government agencies, this URL is not available.

The *dc_description*, *foi_decisionText*, *foi_isAdjoined*, *foi_requestText*, *foi_requestDate*, *foi_requester* and *foi_valuation* attributes are all fields that are published by some suppliers, but not by all, and also not (easily) extractable from the dossier contents. The *foi_publishedDate* and *foi_decisionDate* are available for most suppliers, but when agencies submit directly to us, they are not strictly required, and thus missing for some dossiers. For the attributes in Table 4.3, the *dc_format*, *dc_source* and *dc_title* attributes have missing values for more than five percent of their attributes. As with the

4. A Collection of FAIR Dutch Freedom of Information Act Documents

dossiers, this information is not always supplied by suppliers, and automatic extraction is complicated and not very reliable. For the pages in Table 4.4, the *foi_bodyText* attribute has missing values as for about a quarter of the data there is simply no machine-readable text in the document, possibly a result of scanning. All the other missing values in the table are due to the fact that the PSS- and redacted text detection algorithms are rather expensive to run, and were only included in the processing pipeline in mid 2023.

4.5 Usage Notes

Apart from the dataframes with the dataset, the repository also contains several notebooks that provide guidance on loading the data and working with the dataset. The *Woogledataset.ipynb* notebook is the most important, as it contains instructions on how to load in the datasets, and shows several examples on how to work with the data, as well as containing the code used to generate the numbers and plots in this chapter. The dataset described in this chapter is a static version of the contents of the Woogled search engine, which is continuously updated with new documents. This static dataset has version number 4, which can be used to access the version of the dataset associated with this chapter. Since the number of documents collected is constantly increasing, new versions of the dataset will be made available through DANS in the future.

4.6 Code Availability

For the collection of the dataset, web-scrappers coded in Python were used, together with code to periodically fetch data from APIs, the code is available on Github.¹⁶ The code used to perform the PSS segmentation and redacted text detection is available on GitHub.^{17,18} The usage of other third-party software used in the creation of the dataset has been detailed in the Methods section.

4.7 Conclusion

In this chapter, we aimed to investigate the feasibility of creating a large, searchable collection of Dutch FOIA documents, using a strict metadata scheme and automatic processing techniques, including the ones described in Chapters 2 and 3. To this extent, we created the Woogled dataset, collected through web-scrappers, from over a thousand suppliers of FOIA documents. We standardized the structure of the collection into dossiers, documents, and pages, with a uniform set of metadata associated with each record. We performed OCR to make the documents have machine-readable text and be searchable, and we performed page stream segmentation and redacted text detection to further enhance the quality of the collection. By using these techniques, we have shown that, with some effort, it is possible to create a large searchable collection of

¹⁶<https://github.com/wooverheid/WoogledDocumentatie>

¹⁷<https://github.com/irlabamsterdam/OpenPSSbenchmark>

¹⁸<https://github.com/irlabamsterdam/TPDLTextRedaction>

FOIA documents with mostly off-the-shelf methods, and that such a collection can be a valuable resource for research.

Part II

Evaluation of Extreme Document Segmentation and Clustering

Elements Like Me: BCubed Revisited

The task of clustering is well known in the field of Machine Learning and has many applications within different disciplines. This has also resulted in a variety of different evaluation metrics being proposed for the task. Unlike the task of classification, where metrics such as precision, recall, and F1 are ubiquitous, there is much less consensus within the field of clustering regarding the preferred method to evaluate the performance of clustering methods. In this chapter, we will investigate the BCubed metric [5], a well-known metric for the *extrinsic* evaluation of a clustering, i.e., where labeled data is present. The BCubed metric operates by, for each element in a set of elements, calculating precision and recall scores based on the overlap between the clusters that the element is in, in the reference and hypothesized segmentations.

Although widely used for evaluating clustering algorithms, the BCubed metric has several shortcomings, one of them being that the metric can never become zero. Since this and other qualities are important for evaluation metrics, we aimed to fix these shortcomings and thus proposed the following research question.

RQ4 Can the BCubed metric be repaired in such a way that its shortcomings are addressed while still maintaining its desirable theoretical properties?

We answer this question in the affirmative and propose the ELM metric, which is a modified version of BCubed, which differs in the element-wise calculations used by BCubed, by not including the element itself in these calculations. We prove theoretically that the ELM metric can obtain this zero score, behaves well on degenerate clusterings, and can produce a different ranking of clustering algorithms when compared to BCubed. We continue with an empirical comparison of the two metrics, again showing that ELM can produce rankings of clustering algorithms that are different from those of BCubed.

Finally, we prove that, like BCubed, ELM satisfies a set of constraints posed by Amigó et al. [3] that ‘good’ clustering evaluation metrics should satisfy.

This chapter was published as: R. van Heusden, J. Kamps, and M. Marx. Bcubed revisited: Elements like me. *Discover Computing*, 27(1):5, 2024. doi: 10.1007/s10791-024-09436-7. URL <https://doi.org/10.1007/s10791-024-09436-7>.

5.1 Introduction

We review the external clustering performance metric *BCubed* [5], indicate a flaw and propose a repair. We then evaluate the repair both theoretically and experimentally.

In essence, clustering and (single label) classification perform the same task: given a set of items E , they partition E . However, when it comes to evaluation with comparison to a gold standard, things are very different.

With classification, the number of blocks in the partition is known (the set of labels), and a mapping exists between the true blocks and the predicted blocks (namely the identity mapping on the labels). So, counting errors is straightforward by making the cross table of predicted and gold truth values (the *confusion table*), and computing precision and recall as the diagonal divided by the two margins, respectively.

With clustering, there is (at prediction time) no known number of blocks (as the label set is unknown), and there is no mapping between the predicted blocks and the true labels. This makes counting errors much less straightforward, witnessed by the numerous proposals on how to do this, nicely surveyed and classified by Amigó et al. [3].

The BCubed measure, proposed by Bagga and Baldwin [5], sidesteps the problem of matching true and hypothesized clusters. It does not measure errors over the clusters, but computes a precision and recall value for each element, and then takes the average. I.e., the recall for element e is the fraction of the true cluster of e that is contained in the predicted cluster of e . As each element e is contained in both its true and predicted cluster, both recall and precision of e are always larger than zero, even when a predicted and true clustering are disjoint except for the element e . This can be repaired by leaving out e itself in the calculation of precision and recall of e . In this chapter, we investigate this alternative definition of BCubed (Section 5.2), evaluate the new metric both theoretically and empirically (Section 5.3), and conclude that it retains all positive properties of BCubed, yields a minimum zero score when it should and can produce different rankings for predicted clusterings when compared to BCubed.

5.2 BCubed Revisited

Let E be a set and N^T and N^H two clusterings (partitions) of E , corresponding to the *true* and *hypothesized* clustering, respectively. We use N_e^T to denote the block in N^T containing e , and similarly for N_e^H and N^H . Figure 5.1 shows how precision and recall relative to an element e are defined given the true and hypothesized clusterings N^T and N^H .

The BCubed measure for a given clustering is then the average harmonic mean (the F1-value) of the precision and recall for each element. This F1 value is what is denoted by “BCubed” or “BCubed score” in the literature, a convention also followed in this chapter. This harmonic mean is usually defined as $2PR/(P + R)$, but the equivalent direct definition is insightful here as well. Let $A \oplus B$ denote the symmetric difference of the sets A and B . Then

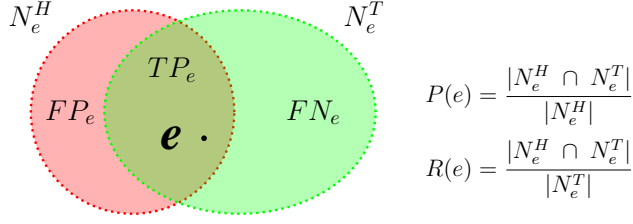


Figure 5.1: Comparing the elements in the true cluster N_e^T of e to those in the predicted cluster N_e^H of e . TP_e , FP_e , and FN_e represent the sets of True Positives, False Positives, and False Negatives for e , respectively. $P(e)$ and $R(e)$ are Precision and Recall relative to e

$$F1(e) = \frac{|N_e^H \cap N_e^T|}{|N_e^H \cap N_e^T| + .5 \cdot |N_e^H \oplus N_e^T|}. \quad (5.1)$$

Figure 5.1 shows that $TP_e \neq \emptyset$, as e is always in TP_e and thus that precision, recall and F1 are always positive for each element, implying that the BCubed score of a clustering is never equal to zero.

Having a meaningful zero point is a requisite for a metric to be measured on ratio-scale. We can say that a score of zero is meaningful if none of the predictions was correct, thus when all items in the contingency table are off the diagonal. Let us formulate this as a desideratum for a clustering metric:

(ZeroScore) For every true clustering, there is a predicted clustering with score 0.

BCubed fails the ZeroScore constraint and can even give quite a high score of .66 to an absolutely wrong prediction. Consider this simple example: $E = \{1, 2\}$ and the true clustering N^T is $\{E\}$. Now let the predicted clustering N^H be (the only other possibility) $\{\{1\}, \{2\}\}$. Obviously it is wrong, but for both elements e , $P(e) = 1$, as it makes no mistakes, $R(e) = .5$, as half of the true elements of the block of e are in its predicted block, and so $F1(e) = .66$. Taking the mean $F1$ over all elements, we get a BCubed score of .66 for this predicted partition.

In fact, because TP_e can never be empty, BCubed fails the ZeroScore constraint in a much stronger manner: for every true clustering there is no prediction with score zero, BCubed can never be equal to zero.

In order to repair BCubed so that it does satisfy the ZeroScore constraint, we only need to remove e from both N_e^H and N_e^T . Thus N_e^T now denotes the set of all elements in the same true cluster as e *except e itself*, and similarly for N_e^H . We call these the *neighbors* of e . Then $TP_e = N_e^H \cap N_e^T$ can be empty, and thus all measures can be equal to zero. The price paid for this is that we may divide by zero in the definitions of P , R and $F1$ and thus must account for that. So all definitions remain the same, but we add the following provisos:

- If $N_e^H = \emptyset$, $P(e) = 1$.

- If $N_e^T = \emptyset$, $R(e) = 1$.
- If $N_e^T = N_e^H = \emptyset$, $F1(e) = 1$.

With these rules the new definitions yield the same scores as the original BCubed definitions on the singleton cases. In the first case, the hypothesized cluster containing e is $\{e\}$, thus no mistakes for e can be made. In the second, recall for e is indeed perfect, and a perfect F1 score for a true singleton is of course only obtained if we exactly predict that.

It is easy to see that with this proviso the definition of $F1(e)$ as in (5.1) is still equivalent to the often used $2PR/(P + R)$ formulation.

5.2.1 A New Name

In the rest of the chapter, we further evaluate this repair. But let us first give it a name. The BCubed measure was introduced by Bagga and Baldwin. In a footnote they attribute the idea of BCubed to Bierman, and thus the cubed Bs. We opted for *ELM*, an abbreviation of *Elements Like Me*, which is a good mnemonic of the way we compute the repaired BCubed measure.

5.2.2 First Impression of the Differences

The following example gives a good impression of the difference between the two measures. Let E be the set consisting of the first 15 digits, and let it have the following true clustering

$$\{\{1, 2\}, \{3, 4, 5\}, \{6, 7\}, \{8\}, \{9\}, \{10, 11, 12\}, \{13, 14\}, \{15\}\}. \quad (5.2)$$

We have generated all possible predictions with the proviso that each cluster must consist of consecutive elements. For a set of N consecutive elements, there are 2^{N-1} of these. For 15 elements, this results in 16,384 possible predicted partitions. Figure 5.2 shows the distribution of the BCubed and ELM scores for all these predictions. Both scores are approximately normally distributed and 25 percent of the ELM scores are below the lowest BCubed score. The ELM scores are more evenly spread over the possible scores. Not only does BCubed start higher, its variance of .006 is much lower than the .02 for ELM. ELM and BCubed can also rank clustering systems differently: in this example, 18% of all ($2^{14} \times 2^{13}$) pairs of predictions are ranked differently by ELM and BCubed.

The Kendall-Tau β statistic (which accounts for tied ranks) between the ELM and the BCubed-based ranking in this example is .63, also indicating that there are substantial differences between the rankings produced by the two metrics.

5.3 Evaluation

We evaluate the new ELM metric both theoretically and empirically in a number of ways:

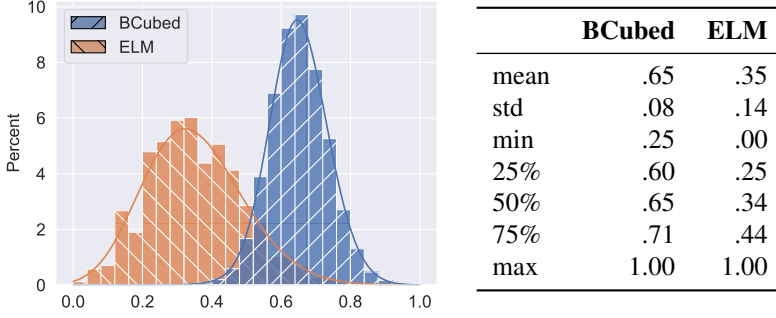


Figure 5.2: Population density diagram of all BCubed and ELM scores for all 16,384 predictions of the model in (5.2) plus the main statistics.

1. Unlike BCubed, ELM satisfies the ZeroScore constraint.
2. ELM has good behaviour on extreme non-informed (referred to as “degenerate” by Beeferman et al. [9]) systems.
3. The ranking of clustering systems based on ELM can be different from the ranking based on BCubed.
4. There are systematic differences between ELM and BCubed in synthetic experiments.
5. There are systematic differences on a real dataset, with a state-of-the-art clustering system based on BERT.
6. ELM satisfies the same four clustering metric constraints developed by Amigó et al. [3] that BCubed satisfies.

We compare the BCubed and ELM versions of P , R and $F1$ using superscripts P_{BCubed} , P_{ELM} , R_{BCubed} , R_{ELM} , $F1_{BCubed}$ and $F1_{ELM}$. In the proofs, the precision, recall and $F1$ scores over a set of elements E are defined as the average of these scores over all elements in E .

5.3.1 ZeroScore constraint

Theorem 1. *For every set E , with at least 2 elements, and a gold standard partition N_T over E , there is a clustering N_H over E such that for every e in E , $F1_{ELM}(e) = 0$.*

Proof. Let E and N^T be as stated in the Theorem. Let $E_s \subseteq E$ be the set of elements which are clustered into singletons. There are three cases: no true singleton clusters, exactly one true singleton cluster or more than one singleton cluster. If there are no true singleton clusters, simply let the predicted clustering partition E into singletons. Recall that we now use N_e^H and N_e^T as denoting all elements in the same cluster as e except e . In particular, with a true singleton cluster $\{e\}$, $N_e^T = \emptyset$. Then for each $e \in E$, $N_e^H = \emptyset$ and $|N_e^T| \geq 1$ (as $|E| \geq 2$). And thus $TP_e = \emptyset$ and $F1(e) = 0$, because the

special clause for $F1$ does not apply. If there is more than one true singleton cluster, create the predicted clustering N^H as follows: one cluster E_s and for each $e \in E \setminus E_s$, a singleton cluster $\{e\}$. Again, we must show that $F1(e) = 0$, for each $e \in E$. First, let $e \in E_s$. Then $N_e^T = \emptyset$ and $N_e^H = E_s \setminus \{e\}$, which is not equal to \emptyset as E has at least 2 elements. And thus $TP_e = \emptyset$ and $F1(e) = 0$ because the special clause for $F1$ does not apply. If $e \notin E_s$, the reasoning is as in the case without singletons. If E_s is itself a singleton, say $\{s\}$, we proceed as follows. Because E has at least two elements, it has another element different from s , say t . Let N^H consist of the cluster $\{s, t\}$ and, again, for each $e \in E \setminus \{s, t\}$, a singleton cluster $\{e\}$. Using the same argument as above, for each $e \in E$, $TP_e = \emptyset$ and $F1(e) = 0$, as the special case never applies. \square

5.3.2 ELM Behaves Well on Degenerate Clusterings

Theorem 2. *Let N^T be a true clustering over a set E and N^H the clustering consisting only of singleton clusters. Then $P(e) = 1$, for all $e \in E$, and $R(e) = F1(e) = 1$ only if $N_e^T = \emptyset$ and 0 otherwise.*

An immediate corollary is that the ELM $F1$ for the degenerate singleton clustering is equal to the proportion of singletons in the gold standard partition.

Proof. Assume E , N^T and N^H are as in the theorem. In particular then $N_e^H = \emptyset$, for all $e \in E$. Then by the special clause in the definition, $P(e) = 1$ for all e , and $R(e) = F1(e) = 1$ if $N_e^T = \emptyset$. When $N_e^T \neq \emptyset$, still $N_e^T \cap N_e^H = \emptyset$, and thus both $R(e)$ and $F1(e)$ are 0. \square

Now consider the other degenerate clustering: all elements are contained in one cluster. Let N^H be this degenerate all in one predicted clustering, with N^T the true clustering over a set E . Then obviously, $R(e) = 1$, for all $e \in E$. Because $N_e^H = E \setminus \{e\}$ and thus $N_e^T \cap N_e^H = N_e^T$, the precision $P(e)$ equals $\frac{|N_e^T|}{|E|} - 1$. And thus the mean precision equals

$$P = \frac{\sum_{e \in E} |N_e^T|}{|E| \cdot (|E| - 1)} = \frac{\sum_{c \in N^T} |c| \cdot (|c| - 1)}{|E| \cdot (|E| - 1)} = \frac{\sum_{c \in N^T} |c|^2 - |c|}{|E|^2 - |E|},$$




where the $c \in N^T$ denote the true clusters. Note that the BCubed mean precision for this degenerate clustering is equal to $\frac{\sum_{c \in N^T} |c|^2}{|E|^2}$. Also note that when we view the clustering as a directed network partitioned into cliques, the ELM precision equals the *density* of this network, which ranges from zero when each clique is a singleton to one only if the network is complete and thus consists of one giant cluster.

We can conclude that for both degenerate clusterings, ELM gives the lowest reasonable score.

5.3.3 ELM Can Produce Different Rankings Compared to BCubed

We give an example of a true clustering and two predicted clusterings (which can be seen as two competing systems), which are ranked differently by ELM compared to BCubed. The clusterings are over the set $E = \{1, 2, 3, 4, 5\}$ and are given in the first

Table 5.1: $F1$ scores per element and the mean, for the given true and two system clusterings over the set $\{1, 2, 3, 4, 5\}$, according to both BCubed and ELM.

$True$							
H_1							
H_2							
Metric	System	1	2	3	4	5	Mean
BCubed	H_1	$\frac{2}{3}$	$\frac{2}{3}$	1	1	1	.87
BCubed	H_2	1	1	$\frac{4}{5}$	$\frac{4}{5}$	$\frac{1}{2}$.82
ELM	H_1	0	0	1	1	1	.60
ELM	H_2	1	1	$\frac{2}{3}$	$\frac{2}{3}$	0	.66

3 rows of Table 5.1, with for example H_1 , depicting the clustering $\{1\}, \{2\}, \{3, 4, 5\}$. The other rows compute $F1(e)$ for each element, for each clustering and using ELM and BCubed. System H_2 is better according to ELM while H_1 is better according to BCubed.

Both H_1 and H_2 contain one error, but the error in H_2 is in the larger cluster. Clustering intuition says that errors in smaller clusters should be penalized more than errors in larger ones, and that is what ELM does here, and BCubed does not.

5.3.4 ELM vs BCubed on synthetic data

We expand on the small synthetic experiment conducted in Section 5.2.2 by computing the BCubed and ELM scores for all clusterings of size 14 against all other clusterings of size 14. As there is a total of 2^{14-1} possible clusterings, we thus have 8,192 experiments, with each of these experiments producing two rankings of the predicted clusters, one for BCubed and one for ELM. The distribution of these scores is shown in Figure 5.3. To further investigate the differences between BCubed and ELM when used to rank systems, we calculate the Kendall-Tau statistic between all rankings and also look at the number of system pairs where the order was swapped between BCubed and ELM (which is part of the calculation of Kendall-Tau). The number of pairs where the ranking order was swapped between BCubed and ELM was roughly 39 billion out of the 274 billion cases (14%). The Kendall-Tau over all pairs of rankings is normally distributed with a mean of .70 and a standard deviation of .06, also similar to the example in Section 5.2.2.

Figure 5.4 shows the distribution of the fraction of the number of swaps for all of the 8,192 experiments. Thus each datapoint is the fraction of possible system pairs where the order between the ranking between BCubed and ELM was swapped for that particular ranking. The y-axis represents all 8,192 rankings and indicates what percentage of all rankings has a certain fraction of swaps. To investigate which type of ground truth clusterings result in the largest number of swaps, we employ the Pearson correlation between the entropy of the ground truth clustering and the number of pairs

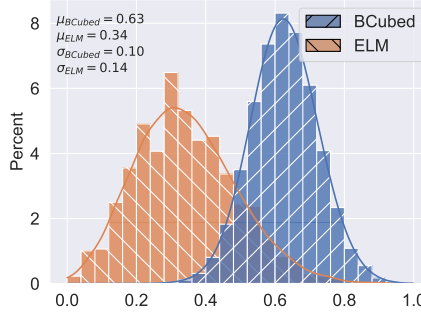


Figure 5.3: Population density diagram of BCubed and ELM scores between all possible pairs of ground truth and predicted clusterings of size 14 plus the main statistics ($N=2^{13} \cdot 2^{12}$).

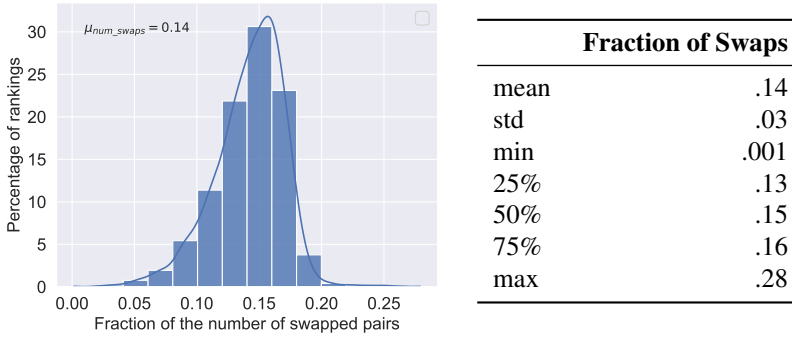


Figure 5.4: The fraction of the number of pairs with reversed orderings between BCubed and ELM for all rankings for all clusterings with size 14 ($N=8,192$) plus the main statistics

swapped in the ranking. The entropy for a given ground truth clustering N^T is given by

$$\text{entropy}(N^T) = - \sum_{C \in N^T} p(C) * \log_2(p(C)) , \text{ where } p(C) = \frac{|C|}{|E|}.$$

The Pearson correlation between the entropy of a ground truth clustering and the number of swaps for that ground truth was .81 ($N = 8,192$). As clusterings that have more small elements have higher entropy, this means that the number of swaps tends to be higher when the ground truth has a larger number of small clusters. This is as expected as the difference between ELM and BCubed is larger on smaller clusters.

5.3.5 ELM Vs. BCubed on Real Data

We compare ELM and BCubed for three fixed cluster-size baselines and a clustering algorithm using BERT [33] on a large dataset consisting of 110 samples (separate

Table 5.2: Mean BCubed and ELM precision, recall and $F1$ scores for the BERT based clustering model evaluated on the Page Stream Segmentation dataset (N=34).

	Precision	Recall	F1
BCubed	$\mu = .93, \sigma = .07$	$\mu = .85, \sigma = .26$	$\mu = .83, \sigma = 0.24$
ELM	$\mu = .92, \sigma = .08$	$\mu = .85, \sigma = .26$	$\mu = .80, \sigma = .24$

clustering problems) with in total 24,180 true clusters over in total 89,491 elements. The mean and median cluster sizes are 4 and 2, respectively. Each sample is a sequence of pages of text divided into documents. Thus each cluster consists of a document, which is a continuous sequence of pages. The elements are thus the pages. This scenario is common in the field of Page Stream Segmentation [121]. On average roughly 35% of the clusters in a stream are singleton clusters.

Following Bagga and Baldwin [5] and Amigó et al. [3], we report the mean average $F1$ scores. Thus for every sample E in our test set, we take the average over the $F1(e)$ for each $e \in E$, and then we take the mean over all samples in the test set.

The dataset, together with the code for all the experiments conducted in this chapter is available on GitHub¹. For the experiments with the BERT model, the dataset was split into a 70% train and 30% test set.

As the fixed page size baselines are not learned, we can use the entire dataset (train and test) for these experiments. The BERT model is evaluated on the test part of the dataset, consisting of 34 samples with 6,347 clusters over 25,676 elements.

We first compare the precision, recall and their harmonic mean for BCubed and ELM on three fixed baselines: the two degenerate clusterings with only singleton clusters and one giant cluster, and a more sensible baseline that evenly partitions a sample into clusters of the mean true cluster size of that sample. The results are shown in Figure 5.5. Note how the plots for precision for the all-singleton prediction and recall for the one-giant-cluster prediction show constant values of 1 for both ELM and BCubed. The plots indicate that the smaller the cluster sizes in the predicted clustering, the larger the difference in both the mean and standard deviation of BCubed and ELM, for all 3 measures, again as expected.

We will now cluster this dataset using the BERT model for Page Stream Segmentation from Guha et al. [46]. In short, this model creates textual representations of each page using a BERT model, and then uses this representation to divide the pages into pages starting a new document and other pages. This classification is equivalent to a clustering. We follow their experimental setup, replacing the English *bert-base* model with the Dutch version² as the dataset is in Dutch. We train the model for 10 epochs, using a batch size of 512 and a learning rate of $2e^{-5}$.

Table 5.2 shows hardly any difference in precision and recall, but still a three percent point difference in $F1$ score. The KDE plots of the differences in Figure 5.6 show the same trend.

This result shows that the differences between ELM and BCubed do not only exist

¹<https://github.com/irlabamsterdam/elm>.

²<https://huggingface.co/GroNLP/bert-base-dutch-cased>

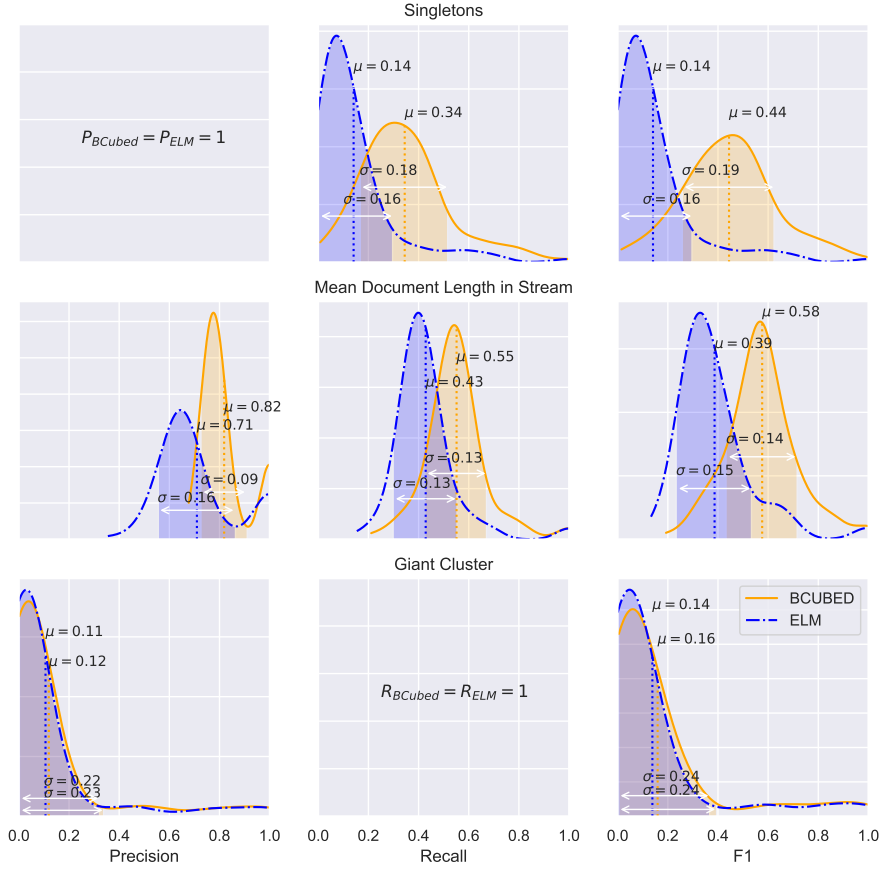


Figure 5.5: Distributions of mean average Precision, Recall and F1 for BCubed and ELM for the three fixed baselines (only singletons, one giant cluster, and each cluster has the length of the samples mean true cluster length (N=110))

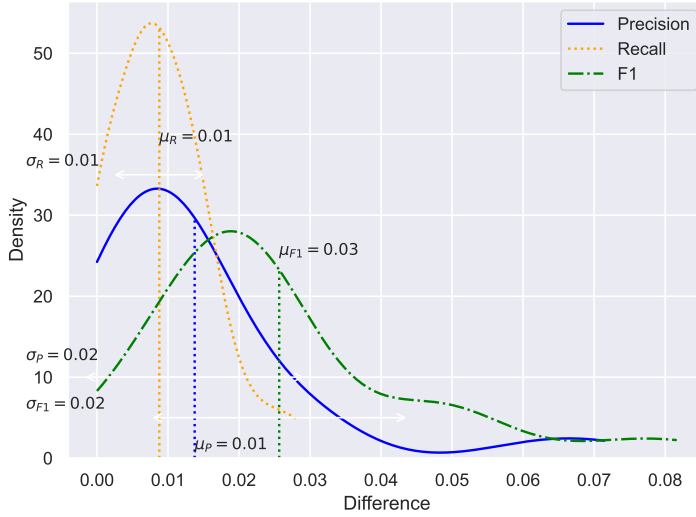


Figure 5.6: KDE Plots of the relative differences between BCubed and ELM for Precision, Recall and F1 for the BERT clustering model (N=34).

on synthetic and simple baseline models, but also on well performing state-of-the-art models tested on large real data.

5.3.6 ELM Satisfies the Constraints of Amigó et al

We show that the four constraints developed by Amigó et al. [3] hold for the ELM $F1$ metric. The family of BCubed-like cluster evaluation metrics is the only one satisfying all these four constraints. For a thorough explanation and motivation of these constraints we refer to the original paper. We follow the same line of reasoning as Amigó et al. [3] and also use their informative pictures.

Homogeneity

The homogeneity constraint states that a cluster assignment D_1 that splits samples into homogeneous subgroups should be scored higher than an assignment D_2 that mixes samples of different subgroups together, like in Figure 5.7.

The ELM recall for each element is the same in D_1 and D_2 , but the precision is lower for the elements in the mixed cluster in D_2 , than in the homogeneous clusters in D_1 . Hence, the mean ELM $F1$ score of D_1 is higher.

Completeness

The cluster completeness constraint states that a cluster assignment D_1 that groups items belonging to the same cluster together should receive a higher score than a clustering D_2 that subdivides items from a homogeneous cluster, like in Figure 5.8.

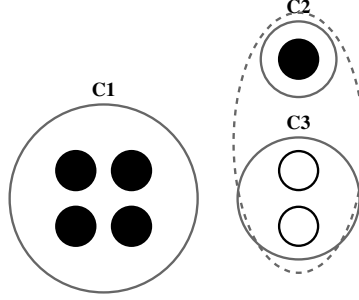


Figure 5.7: Homogeneity constraint: black nodes belong to one cluster and the white nodes belonging to another cluster. Shown are two partitions: the homogeneous $D_1 : \{C_1, C_2, C_3\}$ and the mixed $D_2 : \{C_1, C_2 \cup C_3\}$. Fig. 5.7 is a modification of Figure 5 from Amigó et al. [3].

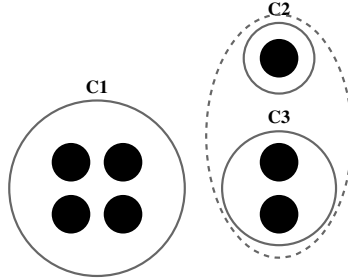


Figure 5.8: Completeness constraint: All nodes belong to the same cluster shown are two partitions: $D_1 = \{C_1, C_2 \cup C_3\}$ and $D_2 = \{C_1, C_2, C_3\}$. Fig. 5.8 is a modification of Figure 6 from Amigó et al. [3].

The argument is the dual of the previous argument. Here, precision is maximal for all elements in both partitions as all clusters are homogeneous. But ELM recall is lowered for those elements in the separate C_2 and C_3 . In fact, recall for ELM is 0 for singleton clusters. Thus the mean ELM $F1$ is higher for the partition D_1 with the joined clusters.

Rag Bag

The Rag Bag constraint states that adding a singleton cluster to a cluster consisting of all differently labeled elements, a *rag-bag*, should score higher than an assignment adding this singleton to a homogeneous cluster, as in Figure 5.9. In this example, this means that D_1 should score higher than D_2 .

First observe that all elements have the same recall in both clusterings. Now the element in C_3 has the same precision of 0 when it is added to C_1 or to C_2 . The elements in the rag-bag C_2 also keep the same precision (namely 0) irrespective to whether C_3 is joined or not. But those in the homogeneous C_1 see a drop in precision (from 1 to $\frac{3}{4}$)

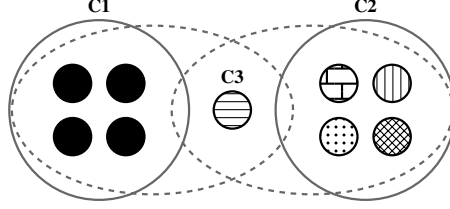


Figure 5.9: Rag Bag constraint: black nodes belong to one cluster and all other nodes are singleton clusters shown are two cluster assignments: $D_1 = \{C1, C2 \cup C3\}$ and $D_2 = \{C1 \cup C3, C2\}$. Fig. 5.9 is taken from Figure 7 from [3].

when C_3 is joined. Thus D_1 has a higher mean ELM $F1$.

Cluster Size vs. Quantity

As stated by Amigo et al., the Cluster Size vs. Quantity constraint can be loosely formulated by saying that small mistakes in large clusters should be penalized less than small mistakes in small clusters. Amigo et al. operationalize this constraint as follows. Let $n > 2$, and E a set of elements with $|E| = 3n + 1$, and let T , H_1 and H_2 be three partitions over E , where T is the ground truth, and H_1 and H_2 are two predicted clusterings. Let T be a partitioning of E containing one cluster C_1 of size $n + 1$, and n clusters each of size 2, C_2 through C_{n+1} . Let H_1 be a partitioning of E that splits C_1 into a cluster C'_1 of size n , and C''_1 of size 1, and with C_2 through C_{n+1} unaltered. Let H_2 be a partitioning that leaves C_1 unaltered, but splits C_2 through C_{n+1} into $2n$ singleton clusters $\{C_2^L, C_2^R, \dots, C_{n+1}^L, C_{n+1}^R\}$. An illustration of this setup for $n = 3$ is given in Figure 5.10. The thus formalized constraint now says that the ELM score of H_1 should be higher than that of H_2 , given T .

Theorem 3 (Cluster Size Vs. Quantity). *Given $n > 2$, T , H_1 and H_2 as described above, the ELM $F1$ score for H_1 is higher than that for H_2 .*

Proof. Let T , H_1 and H_2 be as stated in the constraint for some $n > 2$. Given that both H_1 and H_2 only split true clusters in T into smaller subsets, $P(e) = 1$ for every element in E for both H_1 and H_2 , and thus proving that the mean ELM $F1$ is larger for H_1 than for H_2 simplifies to proving that this holds for the mean recall. We will show that the sum of all $R(e)$ is higher for H_1 than for H_2 , which proves the theorem.

For H_1 , the recall of all $2n$ nodes belonging to the correctly predicted clusters C_2 through C_{n+1} equals 1, and the recall of the single node in C''_1 is 0 (this would be $\frac{1}{n+1}$ for BCubed). The ELM recall of all n nodes in C'_1 equals $\frac{n-1}{n}$ (this would be $\frac{n}{n+1}$ for BCubed). Thus for H_1 , $\sum_{e \in E} R(e)$ equals $2n + n \cdot \frac{n-1}{n} = 3n - 1$.

For H_2 (which correctly predicts the big cluster but splits all true two-size clusters) the ELM recall $R(e) = 0$, for all $e \in C_i$ with $i \neq 1$ (this would be $\frac{1}{2}$ for BCubed). For the $n + 1$ nodes in the correctly predicted C_1 the recall is 1, and thus for H_2 , $\sum_{e \in E} R(e) = n + 1$. For every $n > 1$, $3n - 1 > n + 1$, as desired. \square

















T											
H_1											
H_2											
	1	2	3	4	5	6	7	8	9	10	Mean
H_1	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{2}{3}$	0	1	1	1	1	1	1	$\frac{8}{10}$
H_2	1	1	1	1	0	0	0	0	0	0	$\frac{4}{10}$

Figure 5.10: An illustration of the Cluster Size Vs. Quantity constraint for ELM for $n = 3$ and $E = \{1, 2, \dots, 10\}$. The numbers in the two bottom rows are the ELM $F1$ scores for each element, and the mean $F1$ (the ELM score).

5.4 BCubed in the Literature

We survey for which tasks BCubed has been used and discuss two other refinements of BCubed.

BCubed is used in the Machine Learning community for several clustering problems where a gold standard clustering is available, such as coreference resolution [10, 86, 87, 96, 100], Entity Linking [57, 58], and name disambiguation [4, 38]. In the case of coreference resolution, the task is to map words or short phrases that occur in a text to real-world entities. This mapping defines a clustering of all these words and phrases.

In coreference resolution in particular, BCubed is often used as a successor to the link based metric used in MUC [117]. BCubed has two main advantages over MUC: its ability to score singleton clusters, and the fact that it takes the severity of clustering mistakes into account, something that MUC does not do. ELM obviously still keeps these advantages. In both coreference resolution and Entity Linking, cluster size is likely long tail distributed, with a few very large clusters and numerous smaller clusters, and many singletons. We have seen that BCubed especially overestimates on elements from small clusters and that ELM repairs this. As the reported $F1$ measure is the mean over all elements, this skewed distribution amplifies the overestimation. We thus believe that especially in these applications, ELM is preferable to BCubed.

Several refinements of BCubed have been proposed, to adapt the metric to specific use-cases. Moreno and Dias [81] proposed two adjustments to the BCubed $F1$ metric that makes it more suited for usage with highly unbalanced datasets, which for example occur frequently in the tasks of image clustering, or the clustering of results for ambiguous search terms on the web. They argue that the standard version of BCubed is less suited for this, because the larger clusters (of the irrelevant class) have an unreasonable effect on the total score, comparable to the unreasonableness of accuracy in such cases. Both proposed alterations have the effect of weighting precision more than recall. The most straightforward one is not to use the harmonic mean $F1$, but a differently weighted average. The same remedy can be applied to ELM by using different weights in equation (1) for FP_e and FN_e .

An extension to BCubed that handles overlapping clusters correctly is proposed by Amigó et al. [3], where the quality of a predicted cluster is evaluated by comparing

an element with all other elements (including itself) in the ground truth (for recall, predicted cluster for precision) and comparing how many clusters they share in the prediction compared to the ground truth. However, this extension might assign the maximum F1 score to a clustering that is not exactly equal to the gold standard. Rosales-Méndez and Ramírez-Cruz [90] propose *CICE-BCubed*, which fixes the aforementioned issue for BCubed by also checking for pair occurrences in different classes. The adapted BCubed variant proposed by Amigo et al. that makes it suitable for usage with overlapping clusterings (and the change proposed by the authors of CICE-BCubed), is not straightforward to implement for ELM. The main problem arises from the fact that this extended variant of BCubed must include a comparison between the element and itself, to be able to penalize a model for the spurious creation or deletion of singleton clusters. Consider the example where the ground truth contains two elements e_1 and e_2 that both belong to cluster a , and a prediction where e_1 and e_2 both belong to a , but e_1 also belongs to a new cluster b . Intuitively, the precision for this element should not be 1, as the prediction added a cluster, but the definition of ELM means that this relation is not considered, and thus this mistake is not penalized. We leave the repair of this shortcoming of ELM in the case of overlapping clusters for future work.

5.5 Discussion

We have calculated the F1 scores for both BCubed and ELM on the element level, and then defined the F1 score of a predicted clustering as the average of the F1 scores of all elements. Although we believe this is closest to the original (not explicitly stated) definition as given by Bagga and Baldwin [5], this is not the only way in which BCubed can be defined. Amigó et al. [3] define BCubed from the average precision and recall over all elements and then applying the $2PR/(P + R)$ manner of calculating the F1 score using these averages. In words: we have used the average harmonic mean instead of the harmonic mean of the averages. For the main message of this article this does not matter as in both ways of defining BCubed do not satisfy the ZeroScore constraint.

5.6 Conclusion

In this chapter, we indicated that the BCubed $F1$ measure gives an overestimation of the performance of a clustering method, repaired the definition, and evaluated the result positively.

ELM satisfies a basic property of a metric: it can always obtain the minimal score of zero, and it gives it to each prediction that has nothing correct (i.e., not a single true positive). We want to emphasize that the idea and intuition behind the ELM metric are identical to that of BCubed.

We showed that the difference between ELM and BCubed is largest when the size of true clusters is small and when there are many such small clusters (e.g., when cluster size is power law distributed). Even on large real datasets with a well-performing state-of-the-art clustering algorithm, ELM F1 was three percentage points lower than BCubed.

We end with looking at the problem from the perspective of network science [7, 79]. If we view a clustering not as a set of subsets on some domain D but as a *binary relation on D* , we take a network perspective. A clustering or partition then corresponds to an equivalence relation \equiv . The neighbor function $N(e) = \{e' \in D \mid e \equiv e'\}$ is then the clustering function used to define BCubed and ELM. In network science, it is customary to work with simple (that is, irreflexive), and if possible, undirected relations. If we replace the equivalence relation with this irreflexive undirected relation, we end up with the same partition (in network science, the blocks are called *cliques*). But on this network, the same neighbor function defines ELM, simply because no element is a neighbor of itself. We may speculate how BCubed would have been defined if one of the three B's had been a network scientist.

6

A Sharper Definition of Alignment for Panoptic Quality

After the discussion of the BCubed metric and our proposed alteration ELM, we continue the discussion of evaluation metrics for extreme clustering tasks with the *Panoptic Quality* (PQ) metric, developed by Kirillov et al. [60] for the task of image segmentation and object detection in Computer Vision. The metric operates by creating a one-on-one mapping between predicted and reference objects or segmentations, and then uses a scoring function to evaluate the quality of this mapping. In the original formulation of Panoptic Quality, the mapping between the reference and hypothesized clusterings is created by matching pairs of clusters from the reference and predicted clusterings that have a Jaccard similarity strictly larger than one half. Given non-overlapping clusterings, this constraint guarantees a one-on-one mapping between the reference and hypothesized clusterings. Although this definition ensures a one-on-one mapping, one could wonder if there are any other mappings that also guarantee this one-on-one mapping, and if there is a mathematical definition for these mapping functions. As such, this chapter attempts to answer the following research question.

RQ5 Is there an objective mathematical criterion for defining a matching function that ensures a one-on-one mapping between two sets of (non-overlapping) clusters?

This chapter answers this question in the affirmative, showing that there is indeed a more general mapping that also ensures one-on-one mappings between reference and hypothesized clusterings. We start with a theoretical exploration of this mapping and prove that the mapping is strictly weaker than the mapping used in the original definition of the PQ metric. The chapter continues with an empirical comparison of the two metrics on three real-world Computer Vision datasets, showing the practical effects of this altered matching criterion on state-of-the-art Computer Vision models.

6.1 Introduction

Kirillov et al. [60] have developed *Panoptic Quality* (PQ), a metric that can be used to

This chapter was published as: R. van Heusden and M. Marx. A sharper definition of alignment for panoptic quality. *Pattern Recognition Letters*, 185:87–93, 2024. doi: 10.1016/j.patrec.2024.07.005. URL <https://doi.org/10.1016/j.patrec.2024.07.005>.

evaluate image segmentation methods by comparing predicted and true segmentations. PQ is specifically developed for segmentation problems in which *exact matches* are unfeasible and not even needed for successful applications. Although originally developed for the image domain, PQ can also be used for text, and even for any partitioning problem. The only requirement is that there is an underlying set of elements (in images, the pixels, in text segmentation typically tokens) which are *partially partitioned* (i.e., elements are combined into non-overlapping segments, but not all elements need to be assigned to a segment).

PQ makes the well-known F1 measure (the harmonic mean between precision and recall) available for the partial match segmentation setting. This is done by generalizing the definition of True Positives from a set of items to a set of *pairs of matched items*. If this set is a partial bijection (for every predicted segment h there is at most one true segment t and vice-versa), the false positives and false negatives, and thus all contingency table-based metrics are also defined.

Kirillov et al. suggest to match a predicted segment h to a true segment t iff $IoU(h, t) > .5$, where IoU denotes the intersection-over-union operation, defined as $\frac{t \cap h}{t \cup h}$ ¹. They show that this condition guarantees that the resulting matching is a partial bijection. This condition is simple, effectively computable, and interpretable, as desired by Kirillov et al. [60]. However, even though it is very natural (as it requires that there are strictly more overlapping than missed and spurious pixels), one could ask for a more foundational reason to choose this threshold. This led to the following research question.

Are there other useful², interpretable, simple, and effective matching definitions, which imply the partial bijection property? And if so, is there a most general one?

Indeed there is a strictly weaker, most general and thus sufficient and necessary condition. Let h and t be two segments (thus subsets) of the same set. Now $IoU(h, t) > .5$ is equivalent to $|h \cap t| > |h \oplus t|$, where \oplus denotes the symmetric difference between h and t . In turn this is equivalent to

$$|h \cap t| > |h \setminus t| + |t \setminus h|. \quad (\theta^+)$$

The weaker most general matching definition distributes this and requires with each conjunct implying the injectivity of one side of the matching.

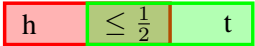
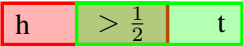
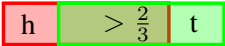
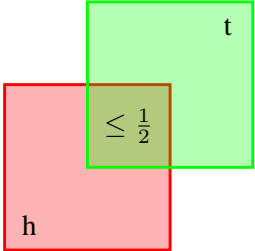
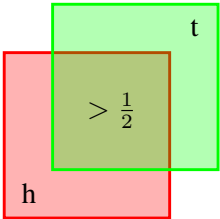
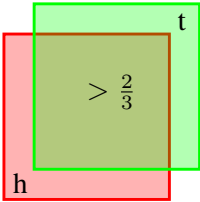
$$|h \cap t| > |h \setminus t| \quad \text{and} \quad |h \cap t| > |t \setminus h|, \quad (\theta^\&)$$

Our main result is that every “fair” matching is a partial bijection if and only if it satisfies $(\theta^\&)$. Below we will develop what “fair” means in this context. Clearly this alternative condition is also simple, natural and interpretable. Table 6.1 shows the difference between $\theta^\&$ and θ^+ .

¹For the definition of IoU used in this chapter, we assume both operands are 2D objects, unless stated otherwise.

²We added the property *useful*, because the identity matching satisfies all other properties, but obviously this is often too strict and thus not (very) useful.

Table 6.1: Examples for three objects of matches with too little overlap for both θ^+ and $\theta^\&$ (left column), only enough overlap for $\theta^\&$ (center column) and enough overlap for both θ^+ and $\theta^\&$ (right column). Green indicates the ground truth object, red indicates the predicted object, \checkmark means the matching satisfies the condition and \times means it fails the condition.

Not enough overlap	Necessary and sufficient overlap	Sufficient overlap
$\theta^\& \times, \theta^+ \times$	$\theta^\& \checkmark, \theta^+ \times$	$\theta^\& \checkmark, \theta^+ \checkmark$
		
		

As a sidenote, non-trivial alternative matching rules exist, such as the one introduced by Chen et al. [23], as an alternative to the θ^+ matching. Instead of enforcing a single threshold that ensures a one-to-one mapping between predictions and ground truth, a mapping is created by using the Hungarian algorithm to obtain an one-on-one mapping without enforcing a single threshold. However, this method still requires an IoU threshold to allow for false positives and false negatives, and is arguably more involved than the θ^+ or $\theta^\&$ matchings.

The rest of the chapter is structured as follows. In the next section we prove this result, and compare the two ways of matching true and predicted objects. We then empirically look at the differences between the two versions of PQ, and finish with related work.

But before we dive into the technicalities, let us compare the two ways of matching from another perspective. The matching defined by $(\theta^\&)$ can equivalently be stated as

$$\frac{|h \cap t|}{|h|} > .5 \text{ and } \frac{|h \cap t|}{|t|} > .5, \quad (\theta^\&)$$

simply stating that the overlap of the two segments should cover more than half of the segments. All objects pairings in the center column of Table 6.1 match by this criterion. They do however not satisfy $IoU(h, t) > .5$, and thus do not match by criterion (θ^+) . In fact, when the two segments have the same size ($|h| = |t|$), then $IoU(h, t) > .5$ is equivalent to asking that the overlap covers more than two-third of the segments, like in the top right image in Table 6.1.

This can be derived from the formula for IoU, where we let $|t \cap h|$ be equal to a , and $|t| = |h| = b$.

$$\frac{a}{a + (b - a) + (b - a)} > \frac{1}{2}$$

Through simple algebraic steps this is equal to requiring $a > \frac{2}{3}b$. We believe this shows that the weaker matching condition is at least as natural as $IoU(h, t) > .5$, but less arbitrary.

6.2 Theoretical results

We first recall the definitions of segmentation, recognition and panoptic quality. Then we show the following basic results about $(\theta^\&)$ and its relation with (θ^+) :

- $(\theta^\&)$ is effectively computable, and defines a partial bijection.
- $(\theta^\&)$ is strictly weaker than (θ^+) .
- Precision, recall, recognition quality and panoptic quality defined with $(\theta^\&)$ are all larger than or equal to the metrics defined with (θ^+) .
- The threshold of .5 is optimal when defining a matching using IoU .

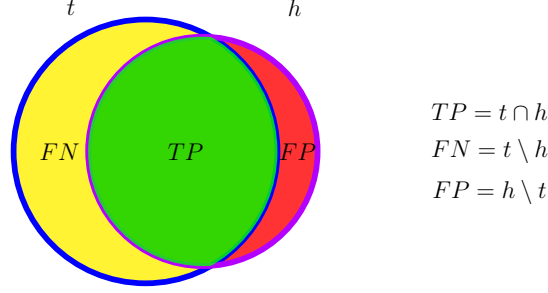


Figure 6.1: Illustration of a ground truth object t and a predicted object h , outlined by blue and purple respectively. The green area represents the overlap, yellow represents the area only in t and red represents the area only in h . TP , FN and FP can then be expressed as set operations on t and h

Then in Section 6.2.1 we prove our main result.

Let E be a set (in images this would be a set of pixels), and both H and T partial partitions of E . Thus both H and T consist of pairwise disjoint subsets of E . We call such H and T segmentations and often ignore the underlying set. Let $b \subseteq H \times T$ be a partial bijection. We will use $\text{dom}(b)$ and $\text{ran}(b)$ to denote the domain and range of b respectively. We often view b as a (partial) function from H into T , and b^{-1} as one from T into H . We call the triple H, T, b an *alignment*.

We recall the definition of the Panoptic Quality metric from [60]. It is given relative to an alignment. Let

$$\begin{aligned} TP &= b \\ FP &= H \setminus \text{dom}(b) \\ FN &= T \setminus \text{ran}(b). \end{aligned}$$

Figure 6.1 contains an example and the way we present these three different parts in this chapter. Now the *recognition quality* RQ is simply the harmonic mean between precision and recall, known as the F1 measure:

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

The Segmentation Quality (SQ) is the mean IoU of the True Positives, and the Panoptic Quality PQ then is the product of SQ and RQ . In a direct definition the relation with $F1$ is even closer. Let wTP denote $\Sigma\{IoU(h, t) \mid (h, t) \in TP\}$. Then

$$PQ = \frac{wTP}{|TP| + .5(|FP| + |FN|)}.$$

This is the reason why one can refer to PQ as a weighted version of $F1$. Similarly, one can define weighted and unweighted versions of precision and recall by dividing wTP or $|TP|$ by $|TP| + |FP|$ for precision and $|TP| + |FN|$ for recall, respectively. Here the metrics are defined relative to one alignment, thus to one example. The PQ , RQ and SQ of a set of examples is simply the mean PQ , RQ and SQ over them, respectively.

Theorem 4. *Let H and T be segmentations of the same set and let $B = \{(h, t) \in H \times T \mid (h, t) \text{ satisfies } (\theta^{\&})\}$. Then B is effectively computable, a partial bijection, and for each $(h, t) \in B$, $\text{IoU}(h, t) > \frac{1}{3}$.*

Proof. A trivial nested for loop over H and T finds the alignment defined by $(\theta^{\&})$. This can be optimized using the order on the elements of the underlying domain. That for each $(h, t) \in B$, $\text{IoU}(h, t) > \frac{1}{3}$, follows from the fact that $|t \cup h|$ is equal to $|t \cap h| + |t \setminus h| + |h \setminus t|$. $(\theta^{\&})$ states that $|t \cap h|$ is strictly larger than these last two summands.

To show that B is a partial bijection, suppose to the contrary that it is not. We will derive a contradiction. There are two possibilities. We treat one: there is a $t \in T$ and two different (and thus disjoint) $h_1, h_2 \in H$ satisfying $(\theta^{\&})$. Thus by assumption, both $\frac{|h_1 \cap t|}{|t|} > .5$ and $\frac{|h_2 \cap t|}{|t|} > .5$. From $\frac{|h_1 \cap t|}{|t|} > .5$ it follows that $|t \setminus h_1| \leq .5|t|$. Because h_1 and h_2 are disjoint, $h_2 \cap t \subseteq t \setminus h_1$, and thus by transitivity, $|h_2 \cap t| \leq .5|t|$, which contradicts our assumption. \square

We now establish the relation between the two ways of matching. We have seen that both (θ^+) and $(\theta^{\&})$ define alignments, and thus give alternative definitions of RQ , SQ and PQ . Given two segmentations H and T of the same set, we speak about their θ^+ - and $\theta^{\&}$ -alignment, and distinguish the corresponding metrics using the same superscripts.

Theorem 5. *$(\theta^{\&})$ is strictly weaker than (θ^+) . That is, every θ^+ -alignment is a $\theta^{\&}$ -alignment, and there are segmentations H, T with an $\theta^{\&}$ -alignment but no θ^+ -alignment.*

Proof. Because $|h \setminus t|$ and $|t \setminus h|$ are disjoint, (θ^+) implies $(\theta^{\&})$. For strictness let $H = \{\{1\}, \{2, 3, 4\}\}$ and $T = \{\{1, 2, 3\}, \{4\}\}$. Then with $(\theta^{\&})$, $\{2, 3, 4\}$ is aligned to $\{1, 2, 3\}$ because it has 2 elements in the overlap, and it has one missing and one spurious element. But the IoU of these two segments is equal to $\frac{2}{4}$ and thus not strictly larger than .5, and thus the alignment defined by (θ^+) is empty. \square

We now investigate what happens to the three panoptic quality metrics when they are defined using $(\theta^{\&})$ and (θ^+) .

Theorem 6. *Consider the θ^+ - and the $\theta^{\&}$ -alignment of the same two segmentations H and T of the same underlying set. Then*

$$\begin{aligned} SQ^{\&} &\leq SQ^+ \\ P^{\&} &\geq P^+ \\ R^{\&} &\geq R^+ \\ RQ^{\&} &\geq RQ^+ \\ PQ^{\&} &\geq PQ^+. \end{aligned}$$

Proof. By the previous theorem, $b^+ \subseteq b^{\&}$, and thus the number of True Positives remains the same or grows with $(\theta^{\&})$, since the alignment condition $\theta^{\&}$ is less strict, and as shown in the example in Theorem 2, allows pairings with an IoU of less than .5

to be considered true positives. The extra true positives have an IoU between $\frac{1}{3}$ and $\frac{1}{2}$, and so these extra TPs will bring the mean IoU , which is SQ , down. Every extra True Positive reduces both the number of FPs and FNs by one. So extra TPs lead to a higher numerator but an equal denominator in the definitions of precision, recall and RQ (which is after all F1) and thus it will go up with the weaker matching condition $\theta^{\&}$. The numerator in the definition of PQ is the sum of all IoU of all TPs. So that also goes up when the number of TPs increases, and thus also $PQ^{\&} \geq PQ^+$. Indeed, the same holds for the weighted versions of Precision and Recall. \square

We will now establish that the IoU threshold of .5 in θ^+ is optimal in the sense that any lower value would not guarantee a partial bijection for all images. We will show this in a more general setting. Both IoU and RQ are instances of a general schema, known as the Tversky index [98]. Given a true and a predicted object t and h and the corresponding TP , FP and FN as defined in the beginning of this section, the Tversky index S is defined as follows.

$$S_{\alpha,\beta}(t, h) = \frac{|TP|}{|TP| + \alpha|FP| + \beta|FN|}, \text{ where } \alpha, \beta \geq 0.$$

The RQ (or $F1$) score corresponds to $\alpha = \beta = .5$, and the IoU to $\alpha = \beta = 1$.

Theorem 7. *Let B denote $\{(h, t) \in H \times T \mid S_{\alpha,\beta}(h, t) > \gamma\}$, for some α, β, γ all between 0 and 1 and H and T segmentations of the same set. Then the following are equivalent.*

1. *B is a partial bijection for all segmentations H and T ;*
2. *both $\frac{\gamma}{1-\gamma}\alpha$ and $\frac{\gamma}{1-\gamma}\beta$ are larger than or equal to 1.*

The theorem immediately implies that B defined by $IoU(h, t) > \gamma$ is guaranteed to be a partial bijection for all segmentations if and only if $\gamma \geq .5$.

Proof. Let B be defined as stated in the theorem. A little algebra shows that $(t, h) \in B$ if and only if

$$|TP| > \frac{\gamma}{1-\gamma}\alpha|FP| + \frac{\gamma}{1-\gamma}\beta|FN|. \quad (6.1)$$

We start with the easy direction. Assuming both $\frac{\gamma}{1-\gamma}\alpha$ and $\frac{\gamma}{1-\gamma}\beta$ are at least 1, (6.1) implies that $|TP| > |FP|$ and $|TP| > |FN|$ and thus $\theta^{\&}(h, t)$ holds and by Theorem 4, B is always a partial bijection.

We prove the other direction by contraposition. Suppose $\frac{\gamma}{1-\gamma}\alpha < 1$. The case for β is shown similarly. We abbreviate $\frac{\gamma}{1-\gamma}\alpha$ by w for ease of notation. Define two total partitions H and T of a set E where $H = \{h\}$ and $T = \{t_1, t_2\}$, satisfying $|t_1| > |t_2| > w|t_1|$. This is possible as $w < 1$. We prove that both (h, t_1) and (h, t_2) are in B and thus B is not a partial bijection. Now $(h, t_1) \in B$ iff (6.1) holds. But we have

- $TP_h^{t_1} = t_1 \cap h = t_1$, as $t_1 \subseteq h$;
- $FP_h^{t_1} = h \setminus t_1 = t_2$, as $t_1 \cup t_2 = h$;

- $FN_h^{t_1} = t_1 \setminus h = \emptyset$, as $t_1 \subseteq h$.

Thus (6.1) reduces to $|t_1| > w|t_2|$, which holds because $w < 1$ and we have constructed t_1 and t_2 such that $|t_1| > |t_2|$. We can similarly show that $(h, t_2) \in B$ using the fact that $|t_2| > w|t_1|$. \square

6.2.1 Every fair alignment satisfies $(\theta^\&)$

We will now prove our main result stating that every reasonable alignment is a partial bijection if and only if it satisfies $(\theta^\&)$. We first develop what are reasonable (we call them "fair") alignments.

Definition 1. Let H, T, b be an alignment.

1. We call $(h, t) \in H \times T$ a *mismatch* in H, T, b if $h \notin \text{dom}(b)$ but $|h \cap t| \geq |b^{-1}(t) \cap t|$.
2. We say that H, T, b is *not fair* if either H, T, b or T, H, b^{-1} contains a mismatch.

Definition 2. 1. The alignment H, T', b' is an *improvement* of the alignment H, T, b if $\text{dom}(b) = \text{dom}(b')$ and $h \cap b'(h) \supseteq h \cap b(h)$ holds for all $h \in \text{dom}(b)$.

2. H', T, b' is an *improvement* of H, T, b if $\text{ran}(b) = \text{ran}(b')$ and $t \cap b'^{-1}(t) \supseteq t \cap b^{-1}(t)$ holds for all $t \in \text{ran}(b)$.

Definition 3. We call an alignment H, T, b *fair* if every improvement of H, T, b is fair.

Thus with an improvement, we may change one of the segmentations and either the range or the domain of the alignment b , but only if the *IoU* of each aligned pair remains the same or increases. Let's see an example:

T	1,2,3		4,5,6
H	1,2,3,4,5		6
H'	1,2,3	4,5	6

We have two alignments H, T, b and H', T, b' , with b and b' as indicated in the mapping. H', T, b' is an improvement of H, T, b as the overlap of the matched pairs remains the same for both segments in $\text{ran}(b)$. The pair $(\{4, 5\}, \{4, 5, 6\})$ is a mismatch of H', T, b' , and thus H', T, b' is not fair. And thus is H, T, b not fair, because it has an unfair improvement.

Theorem 8. Let H and T be two segmentations of the same set and $B \subseteq H \times T$. Then the following are equivalent:

- all $(h, t) \in B$ satisfy $(\theta^\&)$;
- B is a partial bijection and H, T, B is a fair alignment.

Proof (\Downarrow) Assume that all $(h, t) \in B$ satisfy $(\theta^{\&})$. By Claim 2, B is a partial bijection, so we will write B as the function b . Now suppose to the contrary that H, T, b is not a fair alignment. Then there is an improvement of H, T, b which is not fair. There are two cases. We do one, and let the improvement be H, T', b' with an $h \in H$ and a $t \in T'$ such that $|h \cap t| \geq |h \cap b'(h)|$. Because H, T', b' is an improvement, it holds that $|h \cap b'(h)| \geq |h \cap b(h)|$. Thus we have that $|h \cap t| \geq |h \cap b(h)|$.

Now t and $b'(h)$ are disjoint, so $h \setminus b'(h) \supseteq h \cap t$. Because b' is an improvement, $h \cap b'(h) \supseteq h \cap b(h)$ and thus $h \setminus b(h) \supseteq h \setminus b'(h)$, and by transitivity, $h \setminus b(h) \supseteq h \cap t$. Using $|h \cap t| \geq |h \cap b(h)|$ we obtain $|h \setminus b(h)| \geq |h \cap b(h)|$ which contradicts $(\theta^{\&})$.

(\Uparrow) Assume H, T, b is a fair alignment and b a partial bijection. Suppose to the contrary that $(\theta^{\&})$ does not hold. Then one of the two conjuncts fails. Suppose the first. Thus there is an $h \in H$ such that $|h \setminus b(h)| \geq |h \cap b(h)|$. Let $z = h \setminus b(h)$. Now create H, T', b' as follows.

$$T' = \{z\} \cup \{t \in T \mid t \cap z = \emptyset\} \cup \{t \setminus z \mid t \in T \text{ and } t \cap z \neq \emptyset\},$$

and for all $h \in \text{dom}(b)$ set $b'(h) = b(h)$ if $b(h) \in T$, and $b(h) \setminus z$ otherwise.

We will show that H, T', b' is an improvement which is not fair because of h , our required contradiction. Because T and b have these properties, also T' is a partial partition, and b' a partial bijection. To show that H, T', b' is an improvement of H, T, b , we must show that for all $\bar{h} \in H$, the overlap with its match remained the same or increased, i.e., $\bar{h} \cap b'(\bar{h}) \supseteq \bar{h} \cap b(\bar{h})$. This holds by definition of b' when $b(\bar{h})$ and z do not overlap. Thus in particular for the segment h . If they do overlap, then as $z = h \setminus b(h) \subseteq h$, for all $\bar{h} \neq h$, the overlap will increase because the elements in z are disjoint from \bar{h} and thus taking them out of $b(\bar{h})$ reduces the number of errors.

Now we show that H, T', b' is not fair for h , precisely because of the set $z \in T'$ which is not in $\text{ran}(b')$. By definition $z = h \setminus b(h)$ and thus $z = h \cap z$. By assumption on h , $|h \setminus b(h)| \geq |h \cap b(h)|$, and thus $|h \cap z| \geq |h \setminus b(h)|$. We found our desired contradiction.

6.3 Empirical Evaluation

In this section, θ^+ and $\theta^{\&}$ are compared on three instance segmentation benchmarks and their differences evaluated: How many additional True Positives does $\theta^{\&}$ yield? What is their IoU? Are certain classes or visual properties over-represented in the additional TPs? And are they indeed acceptable as correct predictions?

In the experiments, PQ is calculated over the prediction and ground truth sets for all three datasets, where only the matching condition is altered to be either θ^+ or $\theta^{\&}$. Recall that, since the $\theta^{\&}$ matching is less strict than the θ^+ matching, ground truth and predicted pairs with an *IoU* of less than .5 can be considered true positives under $\theta^{\&}$ but not under θ^+ , and thus the number of true positive matchings under $\theta^{\&}$ will be larger or equal to that of θ^+ .

Figure 6.2 shows two examples of additional true positives yielded by $\theta^{\&}$. In the top image, the model makes a recall error and misses a substantial portion of the ground truth object, indicated by the large yellow region. In the bottom image, the model makes a precision error and erroneously predicts a large portion of the image to belong to the



(a) **Partial**: The predicted object only contains a part of the ground truth object

$|t \cap h| = 24023$
 $|t - h| = 23100$
 $|h - t| = 2009$
 $|h \cup t| = 49132$
 $\text{IoU}(t, h) = 0.49$



(b) **Extra**: The predicted object is larger than the ground truth object, and the spurious pixels are not assigned to another object

$|t \cap h| = 4967$
 $|t - h| = 317$
 $|h - t| = 4815$
 $|h \cup t| = 10099$
 $\text{IoU}(t, h) = 0.49$

Figure 6.2: Two examples of TPs according to $\theta^\&$, but not according to θ^+ . The blue and purple contours refer to t and h , respectively. The green area indicates the intersection between t and h , red signifies pixels only present in h , and yellow signifies pixels only present in t . Cardinalities of sets denote number of pixels

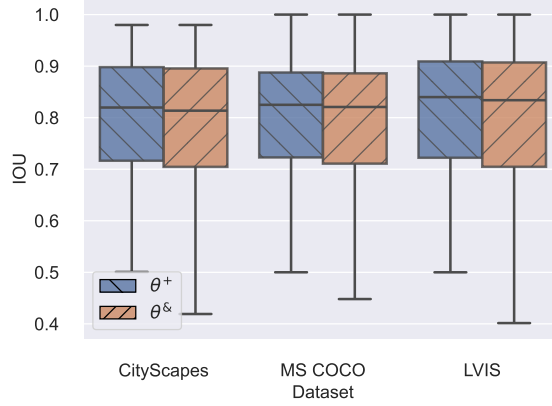
ground truth object, indicated by the red region. In both cases, the IoU equals .49, and thus not enough to be a TP according to θ^+ .

For the comparison, the Mask R-CNN [52] model was run on the CityScapes [29], MS COCO [71] and LVIS(version 0.5) [47] datasets. CityScapes consists of images from streets in several German cities, and MS COCO and LVIS both consist of Flickr images. As is common practice, we use the validation sets to evaluate the performance of the image detection method. It should be noted that CityScapes has much more objects per image annotated ($\mu = 20.4$) than the two other sets. We used the pre-trained Mask R-CNN models made available by Meta through their Detectron2 library [124] for all three datasets. The code and data to reproduce the results of the empirical evaluation are publicly available on Github.³ The main findings can be summarized as follows.

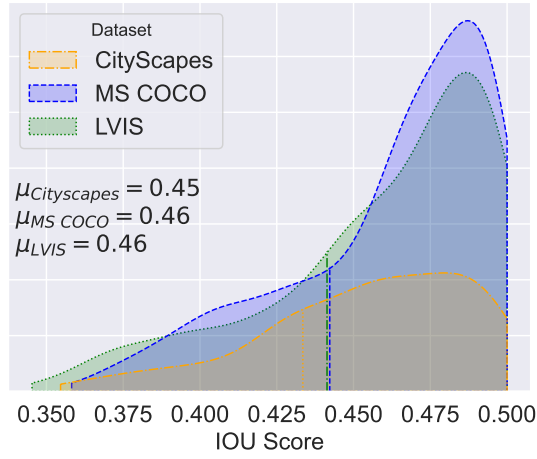
- Evaluating by $\theta^\&$ instead of θ^+ yielded 1,250 additional TPs on the three datasets in total. Per dataset, this meant one to two percent point higher recall.
- The IoU of the additional TPs is close to the .5 threshold of θ^+ ($\mu = .46$, $\sigma = .03$).
- The number of objects in an image and the additional number of true positives in an image are positively correlated, with a Pearson correlation of .35.
- additional TPs are heterogeneous in size, their visual properties and their missed and spurious pixels.
- Five percent of the additional TPs can be considered as being incorrectly classified as detected objects. The majority of these *false hits* are very small objects.

We first describe the distribution of the IoU values of the additional TPs. Table 6.2 shows that $\theta^\&$ yields 1,250 additional true positives counted over all three

³<https://github.com/RubenvanHeusden/GeneralizedPanopticQuality>



(a)



(b)

Figure 6.3: (a): Boxplot of the distributions of the IoU of true positives under both θ^+ and $\theta^\&$ and (b): KDE plot of the IoU of the additional TPs with the vertical lines representing the first quartile.

Table 6.2: Number of additional TPs and the fraction of the additional TPs within all TPs.

Dataset	Additional TPs	Fraction
CityScapes	232	.02
MS COCO	536	.01
LVIS	492	.01

datasets, between one and two percent more true positives than θ^+ depending on the dataset. Figure 6.3a shows the distributions of the IoU for both θ^+ and $\theta^\&$ for all three datasets. For all three, the distributions differ significantly when measured using the Kolmogorov-Smirnov test ($D_{6944,7166} = .03, p = .001$, $D_{18957,19483} = .027, p < .001$ and $D_{14053,14545} = .03, p < .001$), for CityScapes, MS COCO and LVIS respectively, where the subscripts indicate the sample sizes of θ^+ and $\theta^\&$ respectively). Figure 6.3b shows the distributions of the IoU of the additional TPs. For all datasets, over 75 percent of these TPs has an IoU higher than .43.

We now investigate whether the true objects of the additional TPs have particular characteristics. The number of objects in an image and the additional number of true positives in an image are positively correlated ($r(9036) = .35, p < .001$). In all three datasets, the objects are also assigned to classes. The distributions of the classes over all objects and over the additional TPs do not differ significantly for CityScapes and COCO when measured using the Kolmogorov-Smirnov test ($D_{10} = .43, p = .37$, $D_{65} = .03, p = .07$). For the LVIS dataset there is a significant difference ($D_{149} = .23, p < .001$), which can be explained by the fact that the dataset has many, 830, fine-grained labels and that there are only a small amount of additional true positives, so many classes with a few objects in the gold standard do not occur in the additional true positives.

We manually inspected all 232 additional TPs in the CityScapes dataset and classified them into the best fitting error-types, shown in Figures 6.3 and 6.4 ($N_{\text{partial}} = 48$, $N_{\text{extra}} = 45$, $N_{\text{occluded}} = 24$, $N_{\text{crowd}} = 48$, $N_{\text{small}} = 45$, $N_{\text{annotation error}} = 12$).

The classification of error types is partly inspired by previous work by for example Bernhard et al. [11], with similar definitions for the *partial* and *extra* classes, with the work by Bernhard et al. also providing error categories for cases where the predicted- and ground truth objects are disjoint. For all error types, the mean IoU is close to .5, meaning there is not one error type that accounts for most of the low-IoU TPs.

We now investigate whether the additional TPs can really be considered True Positives. The $\theta^\&$ criterion can yield true positive pairs with an IoU of just over one-third, in which case the overlap is only a little bit more than both the missed part and the wrongly assigned part, like in the middle column of Table 6.1. Figure 6.3b shows that in all three datasets there are TPs with an IoU just over a third. If this wrongly assigned part (the FP pixels) overlaps almost completely with another true object, we consider the TP a *false hit*. Figure 6.4b contains an example of such a false hit: the wrongly assigned yellow and red parts are almost as large as the green overlap and the red part is for a large part contained in another object. We make this notion of a false hit precise as



(a) **Partially Occluded:** The ground truth object is partially occluded by another object (b) **Crowd:** The predicted object includes parts of other objects close to it.

Figure 6.4: Examples of the *Partially occluded* and *Crowd* classes

follows, using a parameter $0 < \pi < 1$ to capture "almost as large". A TP (t, h) is a *false hit* if there is a true object $t' \neq t$ such that $\pi \cdot |t \cap h| \leq |t' \cap h|$ and $\pi \cdot |t \cap h| \leq |t \setminus h|$. Note that the first conjunct also implies that $\pi \cdot |t \cap h| \leq |h \setminus t|$ because t and t' are disjoint. It is easy to see that only additional TPs can be false hits. We set $\pi = .75$. Only 61 of the 1,250 additional TPs are false hits, (.5%, 5% and 6% for CityScapes, COCO and LVIS, respectively). These false hits tend to be small objects (within the COCO dataset an object is called small if it has less than 1024 pixels). For CityScapes, MS COCO and LVIS the median pixel size of the true object of the false hits is 359, 422 and 930 pixels, respectively. For CityScapes, MS COCO and LVIS, 100, 71 and 65 percent of the false hits are small objects. Foucart et al. [40] discuss the PQ metric for medical imaging for cell nuclei, and argue against the usage of PQ for small objects, as the small size of the objects means small perturbations in the predictions can have a large effect on whether or not a cell is considered a True Positive. Since the majority of the additional True Positives can be considered correct, we believe that using the $\theta^{\&}$ alignment condition thus yields a more accurate way of measuring model performance.

6.4 Related Work

Traditionally, the performance of image segmentation models has been evaluated using a variety of pixel- and object-level metrics, such as pixel-level precision and recall, and object-level metrics such as Average Precision and average IoU scores [29, 45, 73]. Both Average Precision and average IoU make usage of the *IoU* score to produce a mapping between predicted- and ground truth objects. In particular, Average Precision is calculated by using the *IoU* threshold of .5 and defining precision and recall in a similar manner to PQ. Although these metrics are still widely reported, the emergence of the panoptic segmentation task has resulted in the more widespread adaptation of the PQ metric [26, 52, 61]. Examples include the MaskFormer and OneFormer architectures [26, 52], both of which use the Transformer architecture at the basis of the prediction model, and present unified frameworks for tackling semantic, instance and panoptic segmentation simultaneously. Li et al. provide a detailed overview of the usage of Transformer-based models in their survey paper [70], reporting their results using the PQ metric for panoptic segmentation datasets. Several image segmentation

challenges and benchmarks have also adopted the Panoptic Quality metric as part of their evaluation setup, with examples in diverse fields such as the detection of different types of crops, detection of cell nuclei in the medical domain, the segmentation of 3D point clouds from LiDAR data and other domains such as for modeling attacks on network systems [39, 115, 120]. The PQ metric has also been extended for usage in the video domain, where it is referred to as *Video Panoptic Quality* (VPQ) [59]. In this setting, IoU is used to obtain TP, FP and FN values over a collection of video frames, after which VPQ is calculated in the same way as the original PQ metric. The VPQ metric includes a hyperparameter k that determines how many frames are considered for the calculation of PQ. Cheng et al. [25] propose a change to the calculation of the IoU metric to make it more sensitive to mistakes in object contours. Since the number of pixels around the edge of an object does not scale proportionally with the area of the object, IoU tends to under-penalize boundary mistakes in larger objects. To remedy this problem, they propose *Boundary-IoU*, which only considers pixels up to d pixels away from the object contour and is formalized as follows. Given a distance parameter d , ground truth object $t \in T$ and predicted object $h \in H$.

$$IoU_{Boundary} = \frac{t_d \cap h_d}{t_d \cup h_d} \quad (6.2)$$

Where t_d and t_h are the portions of the ground truth and predicted objects that are up to d pixels away (measured in Euclidean distance) from their respective object contours. The choice of the distance parameter d controls the sensitivity of the metric towards mistakes in the object boundaries: the lower d , the more the object contour decides whether a pair (t, h) is a true positive. Choosing the appropriate value for d is an important consideration, as setting the value of d too low can result in very small perturbations (for example stemming from contour ambiguity) having a large effect on the final score. Cheng et al. experimented with this by comparing the annotations from two annotators on the LVIS dataset, treating one as the gold standard and one as the prediction, varying the value of d and counting the number of TPs. They found that, for the LVIS dataset, setting d to roughly two percent of the image diagonal, and so d would be potentially different from image to image, and could possibly even depend on the ground object size, to avoid the aforementioned problem. For both the VPQ and Boundary-PQ variations, substituting the θ^+ matching by the $\theta^{\&}$ matching still results in partial isomorphism, as both variants constrain the predicted objects to be non-overlapping.

6.5 Discussion & Future Work

Although this chapter explores the effect of the altered alignment condition $\theta^{\&}$ on the PQ metric specifically, the fact that the matching concerns the IoU metric means that the altered condition can be used in any setting where IoU is used in determining the correctness of a prediction against a gold standard. Examples of this include the Average Precision and average IoU metrics discussed previously, as well as metrics in different fields, such as Named Entity Recognition and text segmentation. Depending on the specific application, the fact that the $\theta^{\&}$ matching is less strict than the original formulation

means that in general more True Positives will be yielded with this matching.

If predicted- and ground truth segments are particularly small (such as cell nuclei in medical images), the use of Panoptic Quality as a reliable evaluation metric can become problematic, as outlined by Foucart et al. [40]. As in the *Boundary-IoU* paper, one of the problems is that mistakes around the boundaries of an object are penalized more heavily for small objects compared to larger objects. Experiments on nuclei segmentation datasets in which artificial distortions are applied to the predictions (dilation by 1 pixel, erosion by 1 pixel and 1 pixel vertical shift), show these small perturbations can lead to a significant amount of predictions receiving an IoU of less than .5, in turn resulting in low PQ scores, that do not reflect the actual quality of the predictions when manually inspected. One of the causes of this behaviour is the fact that *IoU* inherently does not weight spurious and mixed pixels equally, with predictions with missed pixels being scored lower than spurious pixels errors of the same size. Although the proposed matching rule would alleviate this problem in some regards as the threshold is relaxed, the inherent problem is still present. Although this problem is usually solved in practice by discarding predictions below a certain size (dependent on the dataset) [1, 126]. Future work can be done in adapting the metric to use cases where small predictions are common. Although this work explores the implications of a new matching rule for Panoptic Quality in the image domain, the practical implication has only been measured for a handful of datasets in the image domain, and the difference between the two matching rules might be more pronounced for different fields such as usage in Named Entity Recognition, something that could also be explored in future work.

6.6 Conclusion

In this chapter, we aimed to find an objective mathematical criterion for defining a matching function that ensures a one-on-one mapping between two sets of (non-overlapping) clusters.

We found a useful, simple, interpretable and effectively computable definition for aligning true and predicted segments which is both a necessary and sufficient condition for the alignment being a partial bijection. If, given a predicted- and true segment, we let TP, FP, and FN stand for the pixels in the overlap, the spurious and the missed pixels, respectively, then the necessary condition aligns the two segments if $|TP| > |FN|$ and $|TP| > |FP|$. This in contrast to the stronger $IoU > .5$ condition which is equivalent to $|TP| > |FN| + |FP|$. The effect of the weaker condition was small but not negligible; on three instance segmentation datasets, it led to a 1-2% increase in recall. Our empirical analysis of these additional true positives shows that 95% of them are indeed valuable, correctly identified objects. The few misses were mostly very small objects. As the new condition is the most general effectively computable alignment that guarantees a partial bijection, we recommend it to be used in future implementations of PQ.

Text Segmentation Metrics: A Survey

As shown in the previous two chapters, there are different types of evaluation metrics for clustering tasks, each with its own strengths and weaknesses. In this chapter, we focus explicitly on the task of text segmentation, which we consider to be a specific instance of clustering. The field of text segmentation is varied, and examples include Named Entity Recognition, article segmentation, and the previously discussed page stream segmentation. To provide an overview of the metrics used for this task and to compare the metrics developed in the previous chapters, we propose the following research question.

RQ6 What is the most appropriate type of metric for the task of text segmentation?

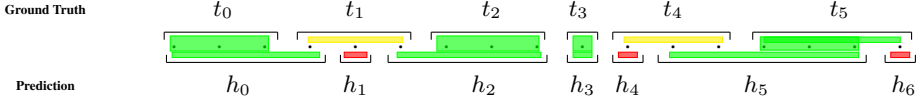
To answer this research question, we compare three groups of evaluation metrics using a set of evaluation criteria, and score each of the metrics using these criteria. We use a combination of theoretical arguments and experiments to examine several metric properties, such as its ability to distinguish between hypothesized segmentations of different qualities and its sensitivity to errors of different severities. Both our theoretical and empirical results show that the group of metrics based on directly comparing segments between the reference and hypothesized segmentations (to which ELM and PQ belong) exhibits the most desirable behavior overall, without the need for specific hyperparameters to control their behavior.

7.1 Introduction

Text segmentation, the task of correctly dividing a text, a sequence of characters or words, into meaningful units, is a necessary preprocessing step in almost every NLP application. Segmentation can be structural (e.g., tokenization, paragraph splitting, chunk parsing) or semantic (e.g., Named Entity Recognition or extractive summarization). The task of text segmentation can be viewed as an instance of clustering, where the clusters (or segments) are non-overlapping, and have to consist of consecutive elements. In addition to merely delineating segments, labeling the segments is often also a part of the segmentation task. For instance, with tokenization, tokens are divided into word

This chapter was published as: R. van Heusden and M. Marx. Text segmentation metrics: A survey, 2025. To be submitted.

7. Text Segmentation Metrics: A Survey



$$TP = \{(t_0, h_0), (t_2, h_2), (t_3, h_3), (t_5, h_5)\} \quad FP = \{h_1, h_4, h_6\} \quad FN = \{t_1, t_4\}$$

$$P = \frac{4}{4+3} = \frac{4}{7} \quad R = \frac{4}{4+2} = \frac{2}{3} \quad F1 = \frac{4}{4+.5(3+2)} = \frac{4}{6\frac{1}{2}}$$

$$\text{Segmentation Quality, } SQ = \frac{\sum_{(t,h) \in TP} \frac{|t \cap h|}{|t \cup h|}}{|TP|} = \frac{\frac{3}{4} + \frac{3}{4} + \frac{1}{6} + \frac{3}{6}}{4} = \frac{3}{4}$$

$$wP = \frac{3}{4} \cdot \frac{4}{7} = \frac{1}{2\frac{1}{3}} \quad wR = \frac{3}{4} \cdot \frac{2}{3} = \frac{1}{2} \quad wF1 = \frac{3}{4} \cdot \frac{4}{6\frac{1}{2}} = \frac{1}{2\frac{1}{6}}$$

Figure 7.1: Reference and hypothesized segments for a typical segmentation task, with seventeen elements that have to be segmented. Green, yellow and red segments indicate True Positives, False Negatives and False Positives, respectively. t_i and h_i refer to the i -th segment in the reference and hypothesized segmentation respectively.

and punctuation tokens, and in traditional NER, entities are labeled with PER, LOC, ORG and MISC. Although these tasks can all be framed as the same text segmentation problem, many of these fields have developed their own evaluation metrics, often dependent on the requirements of systems developed for that particular task. As such, there is a plethora of different evaluation metrics in the literature, each with their own advantages and disadvantages, and without a comprehensive analysis of when these different metrics should and should not be used. In this paper, we survey *extrinsic* evaluation metrics for the task of text segmentation, by which we mean the case where reference and hypothesized segmentations are directly compared (as in Figure 7.1). We aim to provide a comprehensive overview of different metrics and metric groups and, through a series of experiments and argumentation, evaluate the validity of each of the metrics surveyed, providing pointers for users of such metrics to make a more informed choice.

Overlap weighted Precision and Recall. Before we start, let us discuss one (and in fact our favorite) way to do this evaluation. Following Kirillov et al. [60], we use *pairs* of true and hypothesized segments as the True Positives. We say that such a pair is a TP if and only if the overlap is larger (in number of elements) than the spuriously predicted area and larger than the missed area, cf Figure 7.2. This definition ensures that for each true segment there is at most one hypothesized segment, and vice-versa [101]. The set TP can thus be seen as a partial injective function, in other words an alignment. The set FP of False Positives then consists of all predicted segments which do not have a pair in TP, and similarly the set FN consists of all true segments without a predicted pair in TP. With these definitions, precision, recall and the F1 score are defined as usual. For the example in Figure 7.1, precision is $\frac{4}{7}$ and recall equals $\frac{4}{6}$. Kirillov et al.

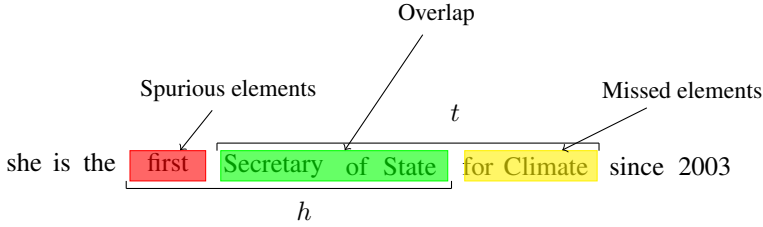


Figure 7.2: A ground truth segment t and an overlapping predicted segment h for a NER task. Missed, overlapping and spurious elements are indicated by yellow, green and red respectively. The pair (t, h) is a True Positive because $|t \cap h| > |t \setminus h|$ and $|t \cap h| > |h \setminus t|$.

[60] proposes to weight these scores with the amount of overlap of the True Positives and uses the Jaccard similarity for that. The average similarity of all True Positives is called the *segmentation quality*, abbreviated as SQ, see Figure 7.1. We then define the weighted versions of precision, recall and F1 by simply multiplying them with this segmentation quality. In Figure 7.1, the SQ is $\frac{3}{4}$, so the weighted precision and recall are reduced to $\frac{3}{7}$ and $\frac{3}{6}$, respectively. The overlap weighted F1 score is called *panoptic quality* in Kirillov et al. [60]. We use the abbreviations wP , wR and $wF1$ for these overlap weighted metrics

The rest of the survey is structured as follows. First we provide a list of the criteria on which we compare the different metrics, and we group the metrics into the three groups that we will evaluate in this paper. We define all metrics in a unified manner, and show them in action using the running example from Figure 7.1. We then compare the metrics using our evaluation criteria and a set of experiments, and draw up conclusions on the advantages and disadvantages of each of the metrics and metric groups. We end with a short discussion of an extension to the PQ metric that makes it usable in settings with overlapping segments.

7.2 Evaluation criteria

Surveys of metrics tend to have a leaderboard-like table with the surveyed metrics on the rows and each column containing a quality indicator by which the metrics are compared. These indicators can be divided into two groups: those that can be formally verified, like *boundedness*, stating that the metric takes values in a bounded interval, and those that can only be argued for, like *scrutiny*, a criterion that concerns the explainability of a metric [80]. Other surveys use the terms objective and subjective for this division, but as rightly pointed out by Moffat [80], the choice of the objective indicators is itself subjective. The metrics surveyed in this paper tend to satisfy the formally verifiable metric properties [3, 80, 84], so we will use these sparsely and concentrate on those properties that need an argument. In more or less their order of importance, these are *meaningfulness*, the already mentioned *scrutiny*, *helpfulness* and *distinctiveness*

- **meaningfulness** Its plausibility as a measurement tool, that is, whether the score it

generates correlates with the underlying behavior it is intended to represent [80].

- **scrutiny/scrutability** The score generated by the metric can be readily explained [80].
- **helpfulness** How useful it is in error analysis and in the improvement of a system [84].
- **distinctiveness** The likelihood of statistically significant system scores being obtained (this concept is discussed in Pevzner and Hearst [84]).

For the first two criteria, *meaningfulness* and *scrutiny*, we will provide arguments on the level of metric groups, and for the *helpfulness* and *distinctiveness* criteria we will conduct experiments on real-world- and synthetic datasets to show adherence of the metrics to these criteria.

7.2.1 The Space of Text Segmentation Metrics

Before providing formal definitions for each of the metrics, we categorize them into three distinct groups, and cover the fundamental assumptions that underlie each of these groups. We make a broad distinction based on whether or not metrics are based on confusion matrices, and on whether performance is measured on the segment- or element level.

Group 1: Segment-Level Metrics

The first group of metrics treat text segmentation as a *segment-level* clustering task, where the segments in the reference and hypothesized segmentations are directly compared in a confusion matrix. The overlap weighted precision and recall metrics defined in the introduction and Figure 7.1 belong to this group. They were originally developed for the evaluation of image segmentation by Kirillov et al. [60].

The Segment Alignment (A) metric from Diaz and Ouyang [34] resembles the segmentation quality SQ as it uses the intersect-ratio and intersection-over-union to create an optimal alignment between reference- and hypothesized segmentations. However, it has no concept of precision and recall, since all segments in the reference- and hypothesized segmentations are matched to a segment in the other segmentation.

The BCubed and ELM metrics [5, 109] compare the segments in the reference- and hypothesized segmentations by calculating precision and recall scores on an element-by-element basis, based on the overlap between the reference- and hypothesized segments that contain the element and taking the average over these scores.

Metrics in this group: wP , wR , $wF1$, A , $BCubed$, ELM

Group 2: Element-Level Metrics

The next group of metrics is also based on confusion matrices, however, instead of making comparisons on a segment-level, segmentations are compared on the level of *elements*. So for instance in the case of a NER classification task, instead of comparing

entities as a whole, individual tokens are being compared. This group includes the well-known precision, recall and F1 measures, as well as metrics based on edit distance, such as the Hamming distance [50], and the S and B metrics introduced in Fournier and Inkpen [42] and Fournier [41]. The latter series of metrics is conceptually similar to precision, recall and F1, but the addition of operators used in the Damerau-Levenshtein distance [31, 67] allows these metrics to assign credit to hypothesized segmentations with small errors in boundary placement. Despite the somewhat involved algorithm required for calculating these metrics, both the S and B metrics have seen some usage in segmentation tasks, such as the segmentation of spoken text and topical segmentation of documents [48, 118].

Metrics in this group: $F1_{Bound}$, $Accuracy_{Bound}$, S , B

Group 3: Metrics Based on a Sliding Window

The third and last group of metrics in this survey is the group that calculates the quality of a predicted segmentation using a *sliding-window* approach. All these metrics use a window of size k (a hyperparameter usually set to half the average segment size) that is slid over the true and hypothesized segmentations simultaneously. For each of these windows, a score is calculated that reflects the agreement of the reference- and hypothesized segmentation over that particular window. The final score of each metric is the mean over the scores for all the individual windows. One of the first metrics in this group is the P_k metric introduced by Beeferman et al. [9]. The P_k metric is derived from a metric which captures "the probability that two sentences drawn randomly from the corpus are correctly identified as belonging to the same document or to different documents". The original metric has several drawbacks, which has led to the development of the *WindowDiff* metric by Pevzner and Hearst [84] which, for each window, yields a boolean value indicating whether the number of segment boundaries in the window is the same for both the reference- and hypothesized segmentations. The chain of improvements of P_k , well described in Scaiano and Inkpen [92], has ended in a proposal, *WinPR*, which also belongs to the this group, but does allow for the calculation of precision, recall and F1 scores, whereas both the P_k and *WindowDiff* metrics do not.

Due to the sliding-window approach, these metrics also allow for the assignment of credit to partially correct solutions with minor inaccuracies in boundary positions. This ability, together with the straightforward implementation of these metrics, has led to this group of metrics somewhat becoming the *de facto* way of evaluating text segmentation, with in particular P_k and *WindowDiff* being used in a significant number of text segmentation tasks [43, 44, 62].

Metrics in this group: P_k , *WindowDiff*, *WinPR*

Other groups

Apart from the three groups described above, there are other metrics that one could use in the text segmentation task, but that we will not cover in this survey. Metrics based

on counting pairs like the Rand index are not suitable because of the quadratic impact, they punish mistakes in large segments much more than those in smaller ones [3]. For clustering, several entropy based metrics have been proposed, but as far as we know they have not been applied to text segmentation. Besides that, Amigó et al. [3] show that they do not satisfy several basic desiderata for clustering metrics.

In this paper, we focus our attention on text segmentation with a single type of boundary. There are several metrics that can be extended to work with boundaries of different types (such as the S and B metrics presented in Fournier and Inkpen [42] and Fournier [41]), however, we consider this scenario to be outside the scope of this survey.

7.3 Formal definitions of text segmentation metrics

In order to provide formal definitions for each of the metrics in this survey, we start by giving a formal definition of the segmentation task itself. Let E be a linearly ordered set, thus in essence a sequence. We use $E[i]$ to denote the i -th element of E (with counting starting at 0). A *segmentation* B of E is a partial partition of E in which all blocks consist of consecutive elements. We will refer to these blocks as *segments*. Thus $B = \{b_0, b_1, \dots, b_l\}$ is a subset of the powerset of E . Note that B can be partial and thus that $\bigcup_{0 \leq i \leq l} b_i$ need not be equal to E . With abuse of notation we say that $e \in B$ if there is a segment $b \in B$ such that $e \in b$. We usually consider a *true* or *reference* segmentation T and a *hypothesized* or *predicted* segmentation H . Segmentations are thus sets of segments.

7.3.1 Segment-level metrics

Following the order of groups that we used before, we start the formalization of the metrics with the group that measures the quality of a predicted segmentation by directly comparing the segments in T and H . For this survey we will discuss overlap weighted precision, recall and F1, A, BCubed and ELM [5, 34, 60, 109].

Overlap weighted precision, recall and F1

These metrics have been defined in the introduction, see Figures 7.1 and 7.2. As these are the most direct confusion table based metrics for the task of segmentation, we use the standard abbreviations without any suffix but prefixed by a w to indicate that they are weighted: wP , wR and $wF1$.¹

BCubed

The BCubed metric (introduced by Bagga and Baldwin [5]) calculates the quality of a hypothesized segmentation by, for each element in the segmentation, measuring overlap

¹Note that these definitions of the unweighted and weighted F1 scores are different from the definitions in Chapters 2 and 3 as in this Chapter the more general matching condition $\theta^{\<}$ from Chapter 6 is used to obtain a partial bijection, instead of the original definition from Kirillov et al. [60].

between the reference- and hypothesized segments that contain the element. For each element $e \in E$, let t_e be $t \in T$ which contains e , if it exists and \emptyset otherwise, and similarly for H and h_e . Assuming non-overlapping segments, this yields a unique segment in t and h for each element e .

Using these definitions of t_e and h_e We can now define element-level precision, recall and F1 scores, as shown in Equations 7.4 and 7.5, where \oplus denotes the symmetric difference between two sets.

$$P_{BCubed}(e) = \frac{|t_e \cap h_e|}{|h_e|} \quad (7.1)$$

$$R_{BCubed}(e) = \frac{|t_e \cap h_e|}{|t_e|} \quad (7.2)$$

$$F1_{BCubed}(e) = \frac{|t_e \cap h_e|}{|t_e \cap h_e| + .5(|t_e \oplus h_e|)} \quad (7.3)$$

Using these element-level definitions of precision, recall and F1, we can now define the BCubed metric over entire segmentations T and H .

$$P_{BCubed}(T, H) = \frac{1}{|E|} \sum_{e \in E} P_{BCubed}(e) \quad (7.4)$$

$$R_{BCubed}(T, H) = \frac{1}{|E|} \sum_{e \in E} R_{BCubed}(e) \quad (7.5)$$

$$F1_{BCubed}(T, H) = \frac{1}{|E|} \sum_{e \in E} F1_{BCubed}(e) \quad (7.6)$$

In the literature, the BCubed measure is the mean F1 value over all elements in E .²

The BCubed metric bypasses the need for aligning hypothesised and true segments by computing something different: for each element e in the domain, it compares the (two unique) hypothesized and true segments containing that element using the standard confusion table metrics. So it computes these metrics relative to that element e : i.e., precision means how many of the elements predicted in the same segment as e are indeed in the true segment of e . Then the mean over all elements is taken.

ELM

In the original definition of BCubed a score of zero can never be achieved if both T and H fully partition E , because the numerator $|h_e \cap t_e|$ in Equations 7.1, 7.2 and 7.3 is always at least 1 because $e \in h_e \cap t_e$. van Heusden et al. [109] propose a slight alteration to the BCubed metric titled Elements Like Me (ELM), that allows the metric to become zero, by removing the element itself from both the h and t segments. In the

²This mean F1 is calculated differently in Amigó et al. [3], where $F1_{BCubed}(T, H)$ is computed directly as the harmonic mean of $P_{BCubed}(T, H)$ and $R_{BCubed}(T, H)$. Because of taking means, this is in general not equal to the mean F1 as defined here.

case of ELM, we alter the definitions of t_e and h_e such that they no longer include the element e itself. Using these altered definitions of t_e and h_e , we can define the precision and recall scores for ELM in a similar manner to BCubed.

$$P_{ELM}(e) = 1 \text{ if } h_e = \emptyset, \text{ otherwise } \frac{|t_e \cap h_e|}{|h_e|} \quad (7.7)$$

$$R_{ELM}(e) = 1 \text{ if } t_e = \emptyset, \text{ otherwise } \frac{|t_e \cap h_e|}{|t_e|} \quad (7.8)$$

$$F1_{ELM}(e) = 1 \text{ if } h_e = \emptyset \text{ and } t_e = \emptyset, \text{ otherwise } \frac{|t_e \cap h_e|}{|t_e \cap h_e| + .5(|t_e \oplus h_e|)} \quad (7.9)$$

Using Equations 7.7, 7.8 and 7.9, precision, recall and F1 over entire segmentations are defined exactly the same as with BCubed.

Segment Alignment (A)

The Segment Alignment (A) metric, introduced by Diaz and Ouyang [34], is similar to the SQ metric, in that it also measures the quality of a predicted segmentation by creating an alignment between reference and hypothesized segmentations. In the A metric, an alignment between two segmentations T and H is created by iterating through both the reference and hypothesized segmentations and, for each segment, selecting the segment from the other segmentation (either reference or hypothesized), that is ‘closest’ to the segment in question, where the distance between segments is calculated using the intersect ratio between two segments, so given a segment t , the closeness to a segment h is defined as $|t \cap h|/|t|$. Note that this way of creating an alignment means that there is not necessarily a one-to-one mapping between segments in T and H , and that a single segment in either segmentation can be linked to multiple segments in the other. This also means that the alignment is not unique, since there could be multiple segments with an equal intersect-ratio with the segment in question. In the case of ties, Jaccard similarity is used to break ties, and if the tie is still not resolved, the leftmost segment is picked as a match.

For given T and H , $A(T, H)$ is the average Jaccard similarity over all pairs in the created alignment between T and H .

7.3.2 Element-level metrics

For certain text segmentation tasks, such as Named Entity Recognition, each word or token in the segmentation should contribute to the final performance of the model, and scores are therefore calculated over all elements in E . Often, there will also be multiple classes in such a segmentation task, where a separate score is calculated for each class, and then aggregated to produce a final score.

For text segmentation tasks with only one type of boundary, it is often desirable to compute the aforementioned metrics only on positive instances of this class, which in the case of text segmentation would be the first elements of the sets in T and H , as these indicate the beginning of a new segment. For $B = \{b_0, \dots, b_n\}$ a set of segments,

\bar{B} denotes the set of all first elements of the segments $\{b_0[0], \dots, b_n[0]\}$. In the case of total partitions, B and \bar{B} define the same set of segments. We can now define precision, recall, F1 and Accuracy on first elements in terms of set operations. Note that $A \oplus B$ denotes the symmetric difference between sets A and B , that is $A \setminus B \cup B \setminus A$.

$$P_{bound}(T, H) = \frac{|\bar{T} \cap \bar{H}|}{|\bar{H}|} \quad (7.10)$$

$$R_{bound}(T, H) = \frac{|\bar{T} \cap \bar{H}|}{|\bar{T}|} \quad (7.11)$$

$$F1_{bound}(T, H) = \frac{|\bar{T} \cap \bar{H}|}{|\bar{T} \cap \bar{H}| + 0.5|\bar{T} \oplus \bar{H}|} \quad (7.12)$$

$$Accuracy_{bound}(T, H) = 1 - \frac{|\bar{T} \oplus \bar{H}|}{|E|} \quad (7.13)$$

Metrics that use the edit distance between the reference- and hypothesized segmentations also measure the quality of a predicted segmentation at the element-level, but often use operators from string similarity metrics to allow for small errors in boundary placement. One of the most straightforward examples of an edit-distance metric is the Hamming distance (Hamming [50]), which simply counts the number of mismatched positions between binary representations of T and H . Usually, this score is normalized by dividing by the length of the sequence, as shown in Equation 7.14.

$$Hamming(T, H) = \frac{|\bar{T} \oplus \bar{H}|}{|E|} \quad (7.14)$$

Thus Accuracy equals 1 minus the Hamming distance. The Hamming distance does not assign credit to an element unless it is completely correct, which is rather stringent, and does not credit boundary placements that are close to the correct boundary. To allow for such partially correct segmentations, the Segmentation Similarity (S) and Boundary Similarity (B) metrics have been proposed [41, 42], both based on the Damerau-Levenshtein distance [67]. These metrics calculate the quality of a predicted segmentation using the *transposition* and *substitution* operators from the Damerau-Levenshtein distance. In these metrics, substitutions are used to account for full misses, where a zero has to be changed into a one in the binary representations, or vice-versa. The transposition operator is used to account for near-misses that are up to k positions apart, where transposing two elements in either the reference or hypothesized segmentation would ‘align’ the elements in both segmentations. Since correcting a shifted boundary now only takes one operation (as opposed to two substitutions), this means the metrics are more ‘forgiving’ in cases of partially correct segmentations. For the formal definitions of both metrics, we start by defining $D(T, H)$, the *minimal* edit distance between T and H , using only transpositions and substitutions. We can write this function D as a combination of transpositions and substitutions. We use \hat{T} and \hat{H} to refer to the binary representations of T and H respectively.

$$D(T, H, w) = SUB(\hat{T}, \hat{H}) + w \cdot TRANS(\hat{T}, \hat{H}) \quad (7.15)$$

Here, w is a parameter that controls how much a transposition is weighted, for further control of the sensitivity to near misses.

Using this definition of the minimal edit distance D , we can define the S metric as follows.

$$S(T, H, w) = 1 - \frac{D(T, H, w)}{|E|} \quad (7.16)$$

Here the edit distance is normalized by the number of elements in E , and this normalized distance is subtracted from 1 to yield a similarity score rather than a distance.

Although the S metric allows for the consideration of small inaccuracies in boundary placement, dividing by $|E|$ is problematic, since this favors segmentations with very few boundaries over segmentations with more boundaries. To address this issue, the authors of the S metric propose Boundary Similarity (B) [41], which changes the normalization constant. Instead of dividing by the total number of elements in E , the B metric divides by the total number of substitutions and transpositions plus the number of correct boundaries, thus excluding True Negatives from the equation.

$$B(T, H, w) = 1 - \frac{D(T, H, w)}{D(T, H, w) + |\bar{T} \cap \bar{H}|} \quad (7.17)$$

As with S , the score is subtracted from 1 to obtain a similarity rather than a distance.

7.3.3 Metrics based on a sliding window

The last group of metrics that we will discuss is the group of metrics that operate by using a sliding window approach. As the name suggests, each of these metrics operates by sliding a fixed-size window over both the reference and hypothesized segmentations simultaneously, and measuring the agreement of both sequences with respect to the presence and quantity of boundaries.

P_k , the first sliding window metric (introduced by Beeferman et al. [9]), is a simplification of the P_D metric introduced in the same paper. The idea behind both these metrics is to measure the *agreement* between a reference and hypothesized segmentation on whether or not two elements are in the same segment. We say that H and T agree on elements $e, e' \in E$ in case e and e' are in the same segment in H precisely when they are in the same segment in T .

The metric $P_k(T, H)$ denotes the chance that H and T do not agree for two elements in E at distance k from each other. We make this formal in Equation 7.18.

$$P_k(T, H) = 1 - \frac{\sum_{i=0}^{|E|+k-2} (\exists t \in T: \{E[i], E[i+k]\} \subseteq t \leftrightarrow \exists h \in H: \{E[i], E[i+k]\} \subseteq h)}{|E|+k-1} \quad (7.18)$$

In the above definition, the sum ranges from 0 to $|E| + (k - 2)$ instead of from zero to $|E| - k$ as in the original formulation. The reason for this is that we pad both ends of the reference- and hypothesized segmentations with $k - 1$ elements (an start counting at zero). In the original definitions of both P_k (and WindowDiff), sliding windows are

calculated over the input elements without padding. Lamprier et al. [65] show that this has undesirable effects, as it means not all elements are included in the same amount of windows. Therefore, they propose padding the input (similar to the padding used for calculating convolutions in Computer Vision). The input is padded by adding $k - 1$ dummy elements to both the start and end of the sequence, where these elements are assumed to be in the same segment as the first or last element respectively.

The WindowDiff metric [84] is an adaptation of P_k that addresses some issues, most notably the fact that, in P_k , both the reference and the hypothesis agreeing on two elements not being in the same segment does not mean that the segmentation is actually correct, as the number of boundaries could still be different. WindowDiff addresses this by explicitly looking at the *number* of boundaries between two elements in the window, as formalized below.

Given a segmentation P , for $e, e' \in P$, let $\delta^P(e, e')$ be a function that returns the number of boundaries in between e and e' , thus δ is always between zero and $|B| - 1$ for a segmentation B .

Similarly to P_k , $\text{WindowDiff}_k(T, H)$ also denotes a chance, but now the chance that H and T differ in the segment distance between 2 elements which are k steps apart in E .

$$\text{WindowDiff}_k(T, H) = \frac{\sum_{i=0}^{|E|+k-2} (\delta^T(E[i], E[i+k]) \neq \delta^H(E[i], E[i+k]))}{|E| + k - 1} \quad (7.19)$$

Scaiano and Inkpen [92] introduce WinPR, which is an adaption of the WindowDiff metric that provides definitions for True Positives, False Positives and False Negatives, allowing for a more detailed analysis of hypothesized segmentations.

Following the notation above, we define TP, FP and FN in the sliding window setting.

$$TP = \sum_{i=0}^{|E|+k-2} \min(\delta^T(E[i], E[i+k]), \delta^H(E[i], E[i+k])) \quad (7.20)$$

$$FP = \sum_{i=0}^{|E|+k-2} \max(0, \delta^H(E[i], E[i+k]) - \delta^T(E[i], E[i+k])) \quad (7.21)$$

$$FN = \sum_{i=0}^{|E|+k-2} \max(0, \delta^T(E[i], E[i+k]) - \delta^H(E[i], E[i+k])) \quad (7.22)$$

In the WinPR calculations, the total number of true positives is dependent on the window size, the number of full misses, and the number of near misses. Here, a full miss is defined as a hypothesized boundary that is more than a distance of k away from the reference boundary. As such, this mistake will be punished k times. For near misses, this depends on the window size, because if the mistake is within the window size, it will only be counted as a mistake $k - d$ times, where d is the distance between the reference-

and hypothesized boundary. In essence, the WinPR metric extends the WindowDiff metric by not only reporting whether or not the number of boundaries is equal, but by measuring the difference in the number of boundaries, and allowing definitions of TP, FP and FN.

WindowDiff and P_k are defined for *total* partitions. Adjusting them to partial partitions means we have to decide how to handle elements in E but not in T or H . We also need to adjust the definition of distance. We do so as follows: For P a partial segmentation of E , let P^* be the smallest total segmentation which contains P . Then each 'gap' in P is filled by one new segment in P^* . We then define distance over these total segmentations, and compute the metrics using all elements in E . In essence we compute the distance and the metrics *as if* all gaps were segments as well, the most natural solution. In the experiments, we will report the sliding-window based metrics as *similarity* metrics rather than distance metrics (so subtracting from 1), to make them easier comparable with the other metrics in this survey.

7.3.4 Metrics on an example segmentation

For clarity, we apply all of the previously discussed metrics to the reference and hypothesized segmentations shown in Figure 7.1, provide derivations for each of the metrics, and show an overview of all scores in Table 7.1.

Segment-Level Metrics

For the calculation of the overlap weighted metrics F1 and $wF1$ we refer to the introduction and Figure 1.

For the calculation of the BCubed and ELM metrics, we can make use of Figure 7.1 to calculate segment overlap between t_e and h_e for each element $e \in E$. Since there is a total of 17 elements in E in the example, we divide by 17 to obtain the final scores for both BCubed and ELM.

$$P_{BCubed}(T, H) = \frac{\left(\frac{3}{4} + \frac{3}{4} + \frac{3}{4} + \frac{1}{4} + \frac{1}{4} + \frac{1}{4} + \frac{3}{4} + \frac{3}{4} + \frac{3}{4} + \frac{1}{4} + \frac{1}{4} + \frac{2}{5} + \frac{2}{5} + \frac{3}{5} + \frac{3}{5} + \frac{3}{5} + \frac{1}{5}\right)}{17} \approx 0.68 \quad (7.23)$$

$$R_{BCubed}(T, H) = \frac{\left(\frac{3}{3} + \frac{3}{3} + \frac{3}{3} + \frac{1}{3} + \frac{1}{3} + \frac{1}{3} + \frac{3}{3} + \frac{3}{3} + \frac{3}{3} + \frac{1}{3} + \frac{1}{3} + \frac{2}{3} + \frac{2}{3} + \frac{3}{4} + \frac{3}{4} + \frac{3}{4} + \frac{1}{4}\right)}{17} \approx 0.72 \quad (7.24)$$

$$F1_{BCubed}(T, H) = \frac{\left(\frac{6}{7} + \frac{6}{7} + \frac{6}{7} + \frac{2}{7} + \frac{1}{2} + \frac{2}{7} + \frac{6}{7} + \frac{6}{7} + \frac{6}{7} + \frac{1}{1} + \frac{1}{2} + \frac{2}{4} + \frac{2}{4} + \frac{2}{3} + \frac{2}{3} + \frac{2}{3} + \frac{2}{5}\right)}{17} \approx 0.65 \quad (7.25)$$

$$P_{ELM}(T, H) = \frac{\left(\frac{2}{3} + \frac{2}{3} + \frac{2}{3} + 0 + \frac{1}{1} + 0 + \frac{2}{3} + \frac{2}{3} + \frac{2}{3} + 1 + 1 + \frac{1}{4} + \frac{1}{4} + \frac{2}{4} + \frac{2}{4} + \frac{2}{4} + 1\right)}{17} \approx 0.59 \quad (7.26)$$

$$R_{ELM}(T, H) = \frac{\left(\frac{2}{2} + \frac{2}{2} + \frac{2}{2} + 0 + 0 + 0 + \frac{2}{2} + \frac{2}{2} + \frac{2}{2} + 1 + 0 + \frac{1}{2} + \frac{1}{2} + \frac{2}{3} + \frac{2}{3} + \frac{2}{3} + 0\right)}{17} \approx 0.59 \quad (7.27)$$

$$F1_{ELM}(T, H) = \frac{\left(\frac{4}{5} + \frac{4}{5} + \frac{4}{5} + 0 + 0 + 0 + \frac{4}{5} + \frac{4}{5} + \frac{4}{5} + 1 + 0 + \frac{1}{3} + \frac{1}{3} + \frac{4}{7} + \frac{4}{7} + \frac{4}{7} + 0\right)}{17} \approx 0.60 \quad (7.28)$$

For the calculation of the A metric we make use of Figure 7.4, which shows the alignment between the reference and hypothesized segmentations, calculated by matching segments from the reference- and hypothesized segmentations with their

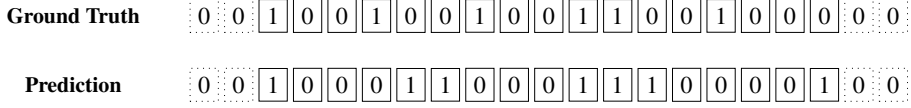


Figure 7.3: Binary representations of the ground truth and predicted segmentations shown in Figure 7.1. We follow the convention of having the first element of both segmentations being a one. The dotted elements at the beginning and end of the sequences are padding elements used for the metrics based on a sliding window. For this example we use a window size of $k = 3$, and therefore we pad with two elements on both the start and end of the segmentations.

closest counterpart based on the intersect-ratio. With this alignment, to calculate the final value of A we simply have to take the average over these values.

$$A(T, H) = \frac{4\frac{1}{4}}{8} \quad (7.29)$$

Element-level metrics

For the calculation of the element-level metrics, we refer to Figure 7.3. There are 3 True Positives, 3 False Negatives, 4 False Positives and 7 True negatives, yielding the following scores.

$$P_{bound}(T, H) = \frac{3}{3+4} = \frac{3}{7}, R_{bound}(T, H) = \frac{3}{3+3} = \frac{1}{2}, F1_{bound}(T, H) = \frac{3}{3+\frac{1}{2}(3+4)} = \frac{6}{13}$$

$$Accuracy_{bound}(T, H) = \frac{3+7}{17} = \frac{10}{17}$$

For the S and B metrics we set the transposition distance to 1, meaning only adjacent elements can be swapped, and we weight these transpositions by .5, thus incurring half the penalty of full misses. Referring to Figure 7.3, there are 3 substitutions and 2 transpositions required to transform the predicted segmentation into the ground truth segmentation. Given that transpositions are weighted by .5, this results in the following edit distance: $D(T, H) = 3 + (.5 \cdot 2) = 4$. There is a total of 3 correct boundaries, and the total number of possible boundaries 17 in the example. We can now calculate the scores for both S and B .

$$S(T, H) = 1 - \frac{4}{17} = \frac{13}{17}, B(T, H) = 1 - \frac{4}{3+4} = \frac{3}{7}$$

Metrics based on a sliding window

For the example in Figure 7.1 we will use a window size of $k = 3$, since taking k to be half the average segment size here would result in k being equal to one, which is not really insightful, and for a window size of two we would be comparing consecutive

7. Text Segmentation Metrics: A Survey

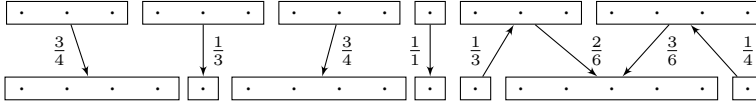


Figure 7.4: Alignment of the segments in Fig 7.1 according to the Maximum Likelihood Assignment from Diaz and Ouyang [34]. The numbers next to the arrows represent the intersection-over-union score of each of the pairs.

Element-Level	Edit-Distance	Segment-Level	Sliding-Window
Accuracy: .59	S: .76 B: .43	SQ: .75 A: .53	P_k : .42 WindowDiff: .47
Boundary		BCubed	ELM WinPR
P: .43		P: .68	P: .59 P: .60
R: .50		R: .72	R: .59 R: .50
F1: .46		F1: .65	F1: .60 F1: .55
		F1	$wF1$
		P: .57	P: .43
		R: .66	R: .50
		F1: .62	F1: .46

Table 7.1: Scores for all the metrics calculated for the example shown in Figure 7.1

elements, which will result in the same scores for all three metrics. With a window size of 3, we will pad with two elements on both the start and the end of the reference- and hypothesized segmentations, copying the value of the nearest element. With 17 elements in the example, and a padding of 3, we have to compute a total of $17 + 2 = 19$ windows

$$P_k(T, H) = \frac{8}{19}, \text{WindowDiff}(T, H) = \frac{9}{19}$$

$$P_{\text{WinPR}}(T, H) = \frac{6}{10}, R_{\text{WinPR}}(T, H) = \frac{6}{12}, F1_{\text{WinPR}}(T, H) = \frac{6}{6 + \frac{1}{2}(6 + 4)} = \frac{6}{11}$$

Using the numbers in Table 7.2 we can compute the P_k , WindowDiff, and WinPR metrics. For the P_k metric, we simply count the number of times that both the reference and the hypothesized segmentation have any boundaries (since there are no instances where they both have none), and divide this by the total number of windows (19 in our case). For WindowDiff we count the number of times the number of boundaries differs between the reference and the hypothesized segments (i.e. the subtraction is not equal to zero).

For the WinPR metric, we have to calculate the number of True Positives, False Positives and False Negatives. In Table 7.2 we can get the number of True Positives by

$bounds_{ref}$	0	0	0	1	1	0	1	1	0	1	2	1	0	1	1	0	0	0	0
$bounds_{hyp}$	0	0	0	0	1	2	1	0	0	1	2	2	1	0	0	0	1	1	0
Agreement	T	T	T	F	T	F	T	F	T	T	T	T	F	F	F	T	F	F	T
$bounds_{ref} - bounds_{hyp}$	0	0	0	1	0	-2	0	1	0	0	0	-1	-1	1	1	0	-1	-1	0
$\min(bounds_{ref}, bounds_{hyp})$	0	0	0	0	0	1	0	1	0	0	1	2	1	0	0	0	0	0	0

Table 7.2: Number of boundaries for all the 19 windows over the reference and hypothesized segmentations in Figure 7.1.

summing the $\min(bounds_{ref}, bounds_{hyp})$ row, yielding 6 True Positives. The number of False Positives is the sum of the negative in the $bounds_{ref} - bounds_{hyp}$ row, and the False Negatives is the sum of the positive numbers, yielding 6 False Positives and 4 False Negatives respectively. With these numbers we can now calculate the precision, recall and F1 scores for the WinPR metric.

7.4 Metric Evaluation

Having formally introduced all of the metrics, in this section we will compare them based on the four criteria we defined earlier, using both theoretical arguments as well as a set of experiments designed to evaluate metric behavior. For the *meaningfulness* and *scrutiny* criteria we will provide a high-level discussion of the different model groups and how well they adhere to these criteria, and for the *helpfulness* and *distinctiveness* metrics we will use a set of experiments on real- and synthetic data to draw comparisons between metrics. All the code and data used to conduct the experiments in this section are available on GitHub ³.

7.4.1 Helpfulness

We start the comparison of the metrics with the discussion of the *helpfulness* criterion. Recall that the *helpfulness* of a system has to do with its ability to aid in the development of effective segmentation systems, as well as its ability to differentiate between systems of different qualities. We assess the ability of the metrics in this regard by conducting two experiments, one focused on the behavior of the metrics on so-called *degenerate* systems, and a second experiment focused on the ability of metrics to differentiate between algorithms of different qualities on a real-world segmentation task.

Performance on degenerate systems

In the first experiment, we test the robustness of the metrics against so-called *degenerate* segmentation models, as discussed by Beeferman et al. [9]. We consider the following three scenarios.

- (*singleton*) a model that predicts every element to be a separate segment.
- (*giant*) a model that predicts all elements to be in a single segment.

³<https://github.com/irlabamsterdam/TextSegmentationMetricSurvey>

7. Text Segmentation Metrics: A Survey

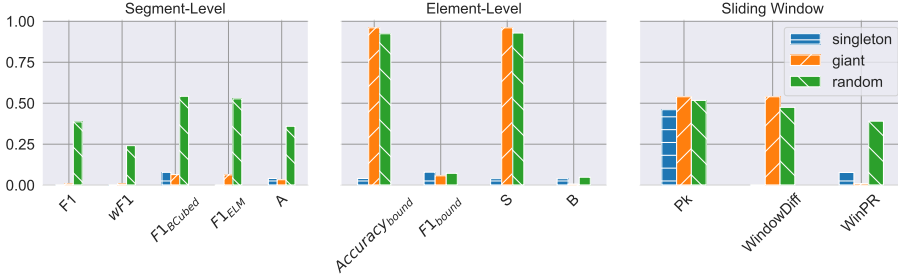


Figure 7.5: Scores of all metrics across the three different model groups on the three degenerate systems, measured on a synthetic dataset (N=1,000).

- (*random*) a model that yields a number of segments equal to the number of segments in the reference segmentation, but these segments are of random sizes.

To evaluate metric behavior under these circumstances, we have generated a synthetic dataset consisting of 1,000 reference segmentations, with an average of 50 segments per segmentation with a standard deviation of 20 segments. Segments have an average size of 25 elements, with a standard deviation of 10 elements. We have chosen these numbers in such a way that the created dataset strikes a balance between segmentations with large segments, and segmentations with smaller segments, as well as having segmentations of different lengths. Experimentation with these parameters resulted in very similar metric behavior across different synthetic datasets.

Figure 7.5 shows the average scores of all metrics on the three degenerate systems, where missing bars indicate a score of zero for that particular system.

In the case of the segment-level metrics, all metrics assign little to no credit to the *singleton* and *giant* systems.

For the element-level metrics, we can see that both the *AccuracyBound* and *S* metrics assign almost perfect scores to the *giant* and *random* systems, but almost no score to the *singleton* system. This is due to the fact that both metrics are calculated by dividing the total number of correct *boundaries* by the total number of boundaries, so if only a few boundaries are predicted, the score will be relatively high, even if these boundaries are incorrect. This is in line with the findings of the authors of the *S* metric, who state that this metric favors sparse segmentations with few boundaries [42]. Note that both the *F1Bound* and *B* metrics do not suffer from this issue, and award low scores to all three systems.

For the metrics based on a sliding window, we can see that the *P_k* metric assigns scores of roughly .5 to each of the degenerate systems, in line with the results of the original paper [9]. *WindowDiff* and *WinPR* both improve on this, with *WindowDiff* reducing the scores of the *singleton* system, and *WinPR* also reducing the score of the *giant* system. Both the *WindowDiff* and *WinPR* metrics require an exact match in the number of boundaries in both reference- and hypothesized segmentations, rather than just an agreement on whether or not the ends of a window are in the same segment, and as such, the score for the *singleton* system is lowered significantly.

The reason that the WinPR metric offers increased robustness to the *giant* system when compared to WindowDiff has to do with the fact that in its calculations WindowDiff also awards credit for True Negatives, whereas WinPR does not.

Note that, except for $F1_{Bound}$ and B, metrics from all three groups assign a relatively high score to the *random* system. This is because the hypothesized segmentation contains the same number of boundaries as the reference segmentation, and therefore will be ‘better’ than the *giant* and *singleton* baselines. In the case of the *giant* and *singleton* baselines, the score a metric gives to these degenerate systems depends heavily on the reference segmentation, and how many boundaries this segmentation contains. In general, when taken over an entire dataset, we expect these scores to be close to zero. In the case of the *random* baseline, the fact that the number of boundaries in the hypothesis is equal to the number of boundaries in the reference segmentation means that there is a somewhat higher probability of correct segmentations. If we have a set of E elements, with b boundaries in both the reference- and hypothesized segmentations, then on average $\frac{b^2}{|E|}$ out of the b boundaries will be placed correctly. We have roughly $25 * 50 = 1,250$ elements per segmentation, with on average 50 segments. This means that on average we expect $\frac{2500}{1250} = 2$ boundaries to be correct, out of total of 50. For the segment-level metrics, the expected number of correct predictions is a bit more difficult to estimate, as it is not directly related to the number of correctly placed boundaries, but rather based on the overlap between reference and hypothesized segments. However, the value of the A metric is in this case a decent estimate of the expected performance, as it measures the degree of overlap (in terms of Intersection-over-Union) between the closest matching segments in the reference and hypothesized segmentations.

Relating the results back to the criterion of helpfulness, we can conclude that, in the case of degenerate baselines, the majority of metrics in the element-level and sliding window-based groups are not helpful, failing to assign appropriate credit to the three baselines evaluated. In the case of the segment-level metrics, all metrics in this group are considerably more helpful in this regard, appropriately scoring the three degenerate baselines.

Performance on a real-world dataset

Although performance on degenerate systems is an important aspect of a metric, it does not reflect the behavior of these metrics in a realistic scenario, on a real-world dataset, and with real-world systems. For this, we will conduct a second experiment on a *page stream segmentation* task, a typical text segmentation task where the goal is to split a stream of pages such that the original document boundaries are retrieved. For this we use the *SHORT* dataset presented in van Heusden et al. [110], which consists of streams of pages of governmental documents that need to be separated into the original documents. We use the predictions of the five text-based models from the paper to evaluate the different metrics. The test portion of the *SHORT* dataset consists of a total of 108 segmentations, with an average of 13 segments per segmentation, and 8 elements per segment.

We will briefly describe each of the models below, for additional details on the training procedure we refer to the original paper [110].

7. Text Segmentation Metrics: A Survey

Segment-Level	F1	0.66	0.67	0.56	0.47	0.55
	wF1	0.62	0.63	0.5	0.42	0.48
	$F1_{BCubed}$	0.78	0.79	0.72	0.67	0.71
	$F1_{ELM}$	0.74	0.75	0.67	0.58	0.65
	A	0.65	0.66	0.56	0.5	0.54
Element-Level	$F1_{bound}$	0.72	0.73	0.6	0.55	0.61
	$Accuracy_{bound}$	0.91	0.91	0.86	0.81	0.87
	S	0.9	0.9	0.87	0.81	0.86
	B	0.58	0.59	0.44	0.4	0.41
	Pk	0.8	0.8	0.75	0.68	0.75
Sliding Window	WindowDiff	0.77	0.77	0.72	0.63	0.72
	WinPR	0.71	0.71	0.6	0.57	0.58
		BERT	TEXT-CNN	KNN Methods	LSTM	XGBOOST

Figure 7.6: Heatmap of the scores of each of the metrics on the *SHORT* dataset from van Heusden et al. [110], where red indicates a higher score, and blue indicates a lower score. Colors are normalized based on the range of each of the metrics separately (N=108)

- **BERT** A classification model based on the BERT [114] architecture, that makes binary classifications on whether a page is a boundary or not.
- **TEXT-CNN** A text classification model based on the architecture described in Wiedemann and Heyer [121], where an LSTM model and a Convolutional Neural Network (CNN) are used for binary classification of boundary pages.
- **LSTM** An LSTM model that, instead of making binary classifications on boundary pages, predicts boundaries for a complete sequence at a time [63].
- **KNN** A K-Nearest Neighbor algorithm that uses TF-IDF vectors of pages to predict segment boundaries.
- **XGBOOST** An algorithm that uses the XGBoost architecture [24] to predict segment boundaries as a binary classification task.

Figure 7.6 shows the scores of each of the five models, where the colors indicate the relative ranking of models (red is higher, blue is lower). What is immediately clear from the figure is that, although the ranges of scores generated by each of the metrics might differ, the relative ranking of the models is similar, with all metrics considering either the BERT or the TEXT-CNN model the best-performing model, with almost identical performance. The same is true for the worst-performing model, with all metrics agreeing that this is the LSTM model.

Although Figure 7.6 shows a high agreement between the different metrics when it comes to ranking models, these are aggregated results, and it is difficult to effectively compare these metrics based on the average scores of the five models alone. To investigate their performance in more detail, we have created Kernel-Density Estimation (KDE) plots in Figure 7.7 for each of the metrics.

As previously mentioned, Figure 7.7 shows that for all metrics, the distributions of the two best performing models, the BERT and TEXT-CNN models, are nearly identical, with only minor differences depending on the specific metric. The differences in metrics are mostly in the distribution of the other three models, and in how these relate to the BERT and TEXT-CNN models.

Starting with the segment-level metrics, we can see that the distributions of the $F1_{BCubed}$ and $F1_{ELM}$ scores are very similar, with the $F1_{ELM}$ metric assigning slightly lower scores in general, which is expected given the altered formulation of the metric. The F1, $wF1$ and A metrics all show a clear separation between the distribution of the two best models, and the three other models.

If we now examine the element-level metrics, we can see that these metrics have much less between-model separation than the segment-level metrics. This is also reflected in the aggregate scores from Figure 7.6, with the KNN and XGBOOST methods achieving scores similar to the BERT and TEXT-CNN methods. The $F1_{Bound}$ and B metrics do not exhibit this behavior, and instead show similar distributions to that of the F1, $wF1$ and A metrics.

Regarding the metrics based on a sliding window, P_k and WindowDiff metrics exhibit nearly identical behavior, where for each of the five models the scores are relatively close together, and all distributions are centered around a specific value. This is much less so for the WinPR metric where, except for the BERT and TEXT-CNN models, the distributions of the other three metrics are much more uniform. This behavior can be somewhat problematic, as it means that simply using the mean value of this metric to rank models can be misleading.

When relating these results to the helpfulness of the metrics, we can conclude that most of the metrics are helpful when ranking real-world systems, and that, although the specific score distributions and value ranges might differ between metrics, almost all metrics rank the five systems in the same order.

Other considerations

Apart from the ability of a metric to behave correctly on degenerate systems, and the ability to properly rank a set of systems on a real-world task, the helpfulness of a metric is also determined by whether or not it allows for a detailed analysis of system performance on the type of errors that are made. Most common are the ability to calculate precision and recall scores, to evaluate if a model is predicting too many or too few segments. For the element-level metrics, only the $F1_{Bound}$ and B metrics have this ability, for the sliding-window based metrics only WinPR can do this, and for the segment-level metrics all metrics except for A can generate precision and recall scores.

To conclude the discussion of the helpfulness criterion, when taking into account behavior on degenerate systems, ranking ability and the ability to produce precision

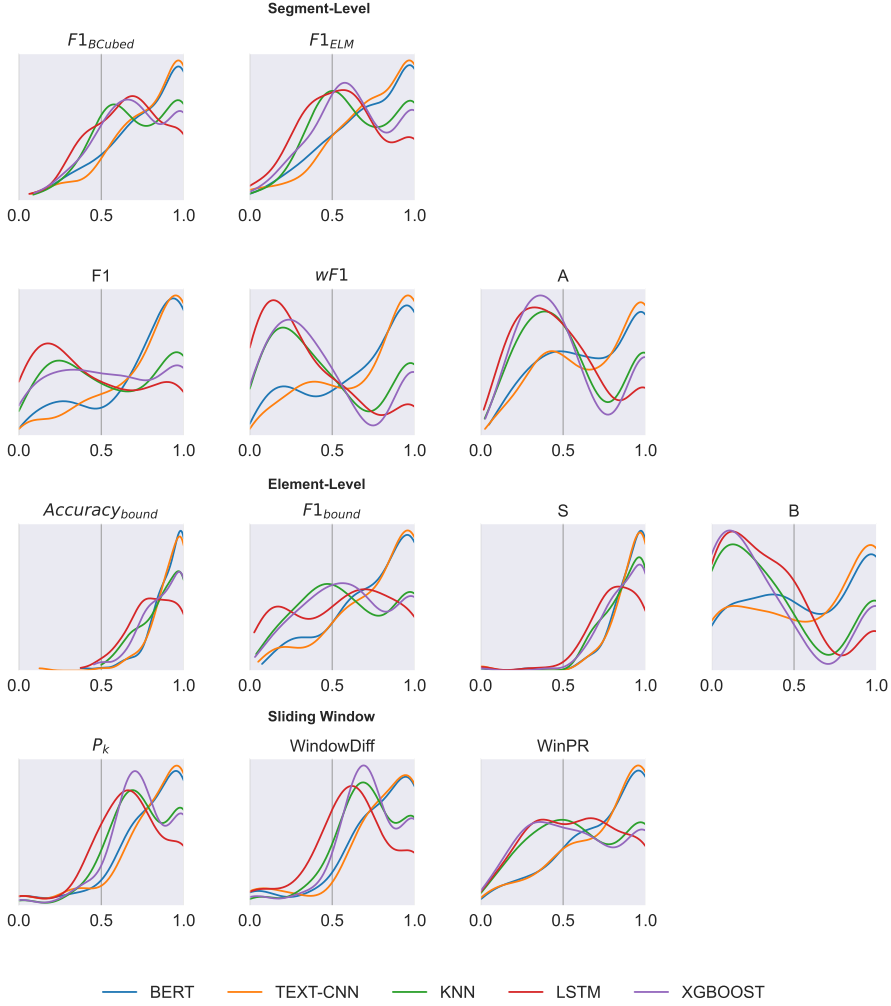


Figure 7.7: Kernel Density Estimation plots of the five segmentation models for each of the metrics on the *SHORT* page stream segmentation dataset (N=108)

en recall scores, we conclude that all metrics in the segment-level group are helpful, that for the element-level metrics only $F1_{Bound}$ and B are helpful, and that for the metrics based on a sliding window, only the WinPRmetric is helpful, albeit somewhat problematic in its generated score distribution as evident from Figure 7.7.

7.4.2 Distinctiveness

We continue with the evaluation of the *distinctiveness* criterion. This criterion states that statistically significant differences in system predictions should result in statistically significant differences in scores. We will measure the distinctiveness of the different metrics using a synthetic dataset, where we artificially introduce boundary shifts into the hypothesized segmentations, and measure the difference in performance under this scenario for the different metrics.

We will use the same generation procedure for the synthetic dataset as before, with 1,000 reference segmentations, with an average of 50 segments per segmentation with a standard deviation of 20 segments. Segments have an average size of 25 elements, with a standard deviation of 10 elements.

We will then randomly move some of the boundaries up to k positions, where we continuously increase k to 20 positions (twice the standard deviation). Note that we set the *maximum* distance for perturbations at k , but that perturbations of less than k can also occur, to mimic a more realistic scenario.

Figure 7.8 shows the results of the experiment on the synthetic dataset, with the distance parameter k on the x axis, and the metric value on the y axis. All of the segment-level metric show a gradual decrease in score as the value of k increases, with some differences in the gradient of the slope, and the point where the scores level off. The slope of the F1 metric deviates somewhat from the other metrics in this group, as it does not directly measure alignment quality. Although more severe errors will cause an increase in False Positives and False Negatives, this is not directly related to the error severity, as we are only counting the number of False Positives and False Negatives, and not their size.

When examining the element-level metrics, it is immediately apparent that both the $Accuracy_{bound}$ and S metrics are not capable of distinguishing between errors of different severities, with both metrics assigning near-perfect scores for all values of k . The opposite is true for the $F1_{bound}$ metric, where shifting boundaries quickly results in scores of almost zero. The fact that we shift *up to* k boundaries means that sometimes boundaries will not be shifted, thus resulting in some credit being given to these segmentations, but it is clear that the $F1_{bound}$ metric also has a very limited capability when it comes to distinguishing between errors of different severities. Although the B metric is somewhat capable of this behavior (the scores do not drop to zero), it is clear that beyond a certain (relatively low) value of k , the metric can no longer make distinctions between error severities.

Since the metrics based on a sliding window have been designed specifically with the ability to assign credit to partially correct segmentations in mind, it is unsurprising that these metrics indeed show a more progressive downwards slope when compared to the element-level metrics. The P_k WindowDiff and WinPR metrics exhibit almost

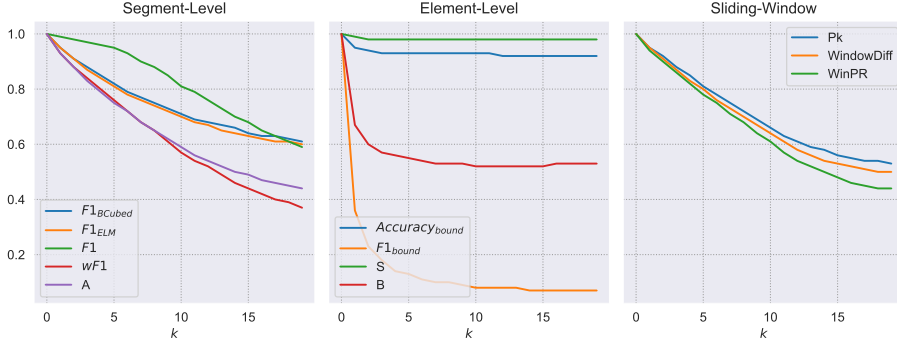


Figure 7.8: Results of randomly moving segment boundaries up to k positions compared to the ground truth on a synthetic dataset ($N=1,000$). The y axis range is shared among all three plots.

identical behavior, with roughly the same degree of degradation when k is increased.

Based on this experiment, we conclude that none of the element-level metrics has sufficient distinctiveness, and that for both the segment-level and sliding window based metric groups, all metrics exhibit sufficient distinctiveness.

7.4.3 Meaningfulness

Recall that the *meaningfulness* criterion states that a metric must be plausible as a measurement tool, and that the generated scores must correlate with the underlying behavior it is intended to represent. In this section we provide arguments for the meaningfulness of each of the different metric groups.

Segment-Level Metrics

Although the BCubed metric (and by extension the ELM metric) operate over all the elements in a segmentation, the approach taken means that they are less prone to the problems faced by the element-level metrics. For each element e in the domain, it compares how many elements in the predicted segment of e are also in the true segment of e (e -precision) and vice-versa (e -recall). Precision and recall for the whole text are then the mean values over all elements. The intuition is that one measures for each element e , how well the elements which are predicted to be *similar to e* (i.e., are in the same segment) are also *truly similar to e* , and vice-versa. The F1, $wF1$ and A metrics, since they operate on the segments directly, again provide increased meaningfulness when compared to the element-level and sliding window based metrics. The scores are, unlike BCubed and ELM, directly computed over segments, and no averages have to be taken over elements. Additionally, they allow for the consideration of non-perfect segmentation, without the need for a hyperparameter, as is the case of the sliding window metrics.

Element-Level Metrics

In the group of the element-level metrics, the task of text segmentation is seen as an element-level classification task. In this scenario, the elements would be characters, words, sentences or other units, and a score is calculated for each individual element. Although intuitively straightforward, this approach has some limitations when applied to text segmentation. Imagine a scenario in which a NER system has predicted every entity perfectly, but all predictions are off by one position. In this case the element level scores for such as system would be very low. However, although the hypothesized segmentation is not correct, the segmentation is still intuitively ‘close’ to the reference segmentation, and thus from a practical point of view should still obtain a reasonably high score. Although the ability of the S and B metrics to handle small mistakes in boundary placements makes them somewhat more meaningful, there are still inherent problems with this approach, mostly based on the fact that they still treat the task of text segmentation as a binary classification task, with little to no concept of segments. This means that these metrics fail to meet several of the formal constraints posted by Amigó et al. [3] regarding clustering metrics.

Sliding Window Metrics

Metrics based on sliding windows are more meaningful than element-level metrics, in that their formulation considers segments when comparing reference- and hypothesized segmentations. As such, they exhibit better behavior on some of the formal constraints as posed by Amigó et al. [3], such as punishing errors in smaller segments more than errors in larger segments, however this behavior is largely dependent on the window size parameter k . This is somewhat problematic, since this value is dataset-specific, and as such comparisons between different datasets are difficult, since it is not clear in advance what kind of impact a specific value of k will have on the behavior of the metric. As such, we also consider these sliding window metric to be of limited meaningfulness in practical scenarios.

7.4.4 Scrutiny

Recall that the scrutiny of a metric has to do with whether or not the scores generated by a metric can be readily explained, simply by observing the scores outputted by the metric. Although the *meaningfulness* of a family of metrics is somewhat uniform among its members, the scrutiny of a metric can vary somewhat per metric, depending on the specific way in which a metric is calculated. Although it is difficult to objectively judge the scrutiny of a metric, we can make some more general statements about the scrutiny of metrics.

Segment-Level Metrics

For the segment-level metrics, the level of scrutiny they exhibit is somewhat similar to the element-level metrics, although the concept of segments makes it a bit more difficult to interpret scores. This is why especially for these types of metrics, precision and recall scores are important, as these can specifically indicate certain model behaviors. As

7. Text Segmentation Metrics: A Survey

	meaningfulness	scrutiny	helpfulness	distinctiveness
Segment-Level	✓	✓	✓	✓
Element-Level	×	✓	×	×
Sliding-Window	×	×	✓	✓

Table 7.3: Overview of the three metrics groups surveyed in this paper with check-marks and crosses indicating whether or not they satisfy each of the four criteria.

such, the A metric has limited scrutiny, as the manner in which the alignment is created (where the alignment is not a bijection, and any-to-many relationships can exist), makes it difficult to interpret this score, and to make conclusions about model behavior. In this group, the F1 and $wF1$ metrics are arguably the most scrutinious, as the subdivision into separate parts by usage of the F1 and $wF1$ metrics mean that multiple aspects of the alignment quality of reference- and hypothesized segmentations can be explained using the metrics. Both the BCubed and ELM metric are somewhat in the middle of this, with the fact that averages have to be taken over elements making the meaning of the scores a bit more obscured, but the precision and recall scores still providing ample signal for evaluation model behavior.

Element-Level Metrics

Due to their simple representation of the segmentation task, the element-level metrics have a high scrutiny, since the calculation of the scores is straightforward, and what is being measured is clear (the number of misaligned boundaries between the reference and hypothesized segmentations). The metrics based on edit-distance are on a similar level when compared to the boundary F1 and accuracy, with the addition of the transposition operation to account for small errors in predictions. The k parameter is rather intuitive, and indicates the distance between different boundaries before they are considered full misses.

Sliding-Window Metrics

In the case of the sliding window based metrics, the P_k and WindowDiff metrics are somewhat scrutinious, as the definition of agreement is relatively easy to understand, but the lack of separate scores for precision and recall, and the influence of the window size make it difficult to easily understand the meaning of a certain score. Take for example the scores of the P_k metric on the degenerate systems, with three completely different types of predictions, and yet almost identical scores. The addition of the calculation of False Positives and False Negatives in WinPR aids somewhat, but again the window-size makes the generated confusion metrics difficult to interpret at first glance.

7.4.5 Overview of Metric Qualities

Having compared all the metrics using the four listed criteria, we now take a step back, aggregate the results of these experiments, and draw some general conclusions about the different metric groups, using Table 7.3 which contains an overview of the three metric groups, and whether or not they meet the listed criteria.

Segment-Level Metrics

Out of all three metric groups, the segment-level metrics show the most desirable behavior on the four evaluation criteria that we posed. Not only do these metrics allow for the proper scoring of models with small boundary inadequacies, they do so without the need for any hyperparameters, which makes them more versatile than the metrics based on a sliding window. The experiments also show that they behave appropriately when presented with degenerate systems, and the presence of precision and recall scores for most of them not only makes them more helpful, it also add to the meaningfulness of these metrics. Although maybe not as scrutinous as some of the aforementioned element-level metrics, their basis in clustering provides a good explainability in the outputs of the metrics nonetheless.

Element-Level Metrics

When examining Table 7.3, and considering the results of the conducted experiments, we can conclude that the metrics that measure the quality of a predicted segmentation at the element-level are poorly equipped to deal with the task of text segmentation. Although the concept of calculating scores at the element level means the metrics have high scrutiny, this approach fails to capture some of the essential qualities of the text segmentation task. This is most visible when examining Figure 7.8, where the inability of these models to handle partially correct predictions, a fundamental quality in text segmentation, is highlighted. Although still regularly used in the literature because of their simplicity, this survey shows that these metrics should be avoided for evaluating text segmentation tasks.

Sliding-Window Metrics

Given that metrics based on a sliding window were created to solve some of the problems with the element-level metrics, it is expected that they improve over these metrics in some respects. Most notably, they allow for the assignment of credit to hypothesized segmentations with small inaccuracies in boundary placement, which results in an increased distinctiveness when compared to the element level metrics. However, there are some significant downsides to this type of approach, mostly notably the need for a parameter k for the window size. Although it allows for controllability in the behavior of the metric, it also means that direct comparisons between datasets are difficult, as this parameter k is dataset dependent, and so the helpfulness of these metrics is somewhat limited. The performance of the P_k and WindowDiff metrics on degenerate systems is also questionable. Although the WinPR metric improves on this, the experiments on a real-world dataset show that the score distribution of WinPR is quite uniform, which

means simply reporting averages for this metric is most likely not informative enough to make proper decisions about model performance.

7.5 Overlapping segmentations

In this section, we examine the case of overlapping segmentations, where an element can belong to multiple classes simultaneously, and show how the overlap weighted segment based precision and recall metrics can naturally be adapted for such a scenario. We will sometimes use the term *clusterings* instead of segmentations because both the problem and our solution occur in the more general case of (partial) clusterings.

In this paper we consider the scenario of overlapping clusters where, at the boundaries of segments, elements can overlap. An example of a fuzzy border in topical text-segmentation is a ‘bridge-sentence’ between two topical sections which could be added to both segments.

In Amigó et al. [3], the BCubed measure is extended to *overlapping clusterings*, by not only considering whether or not two items share a cluster, but by additionally counting how many clusters are shared. This metric has the undesirable property that a hypothesized clustering which is not equal to the reference clustering can still receive a perfect score. This is repaired in [90] by introducing a matching function that attempts to establish mappings between predicted and ground truth clusters. Although this addresses some of the shortcomings of the original formulation from [3], it does require significant alterations to the original metric, and leads to a more complicated formulation.

In the case of segmentations with fuzzy borders, segments can overlap at their boundaries, and thus elements can belong to more than one class. With fuzzy border clustering, both T and H are subsets of the powerset of E , without the non-overlap constraint. In order to define the $wF1$ metric over segmentations with possibly overlapping clusterings, we borrow the concept of cluster *cores* from the k-clique percolation method for graph clustering from Palla et al. [82].

For $t_i \in T$ define the *core* of t_i , denoted by \hat{t}_i , as $t_i \setminus \bigcup_{j \neq i} t_j$, and similarly for $h_i \in H$. By definition, cores do not overlap, and whenever \hat{t} is not empty ($\hat{\cdot}$) is an injective function. We can now define the weighted P, R and F1 scores as follows:

1. compute the set of TPs using the cores;
2. compute the unweighted scores using this set of TPs ;
3. compute the weight SQ as the mean IoU of these TPs but then computed over the original clusters, that is

$$SQ = \frac{\sum_{(\hat{h}, \hat{t}) \in TP} IoU(h, t)}{|TP|}.$$

4. The weighted scores are then the unweighted scores multiplied by the weight SQ , as before.

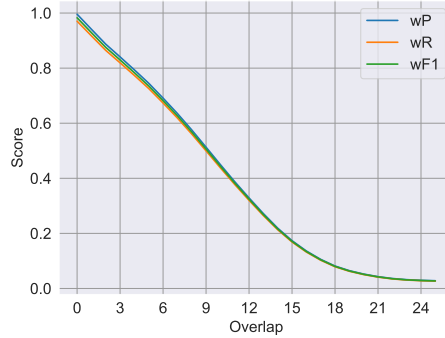


Figure 7.9: Progression of the wP , wR and $wF1$ scores in the case of overlapping clustering, with the amount of overlap between the hypothesized clusters being controlled by the parameter k ($N=1,000$)

Note that, in the case that both the reference- and the hypothesized segmentations contain no overlapping segments, this formulation is equivalent to the original definition, since the cores of the segments are simply the segments themselves.

To investigate the behavior of the thus adjusted $wF1$ metric on segmentations with different grades of overlap, we construct a synthetic dataset to see the impact of various levels of overlap, and artificially increase the amount of overlap in the hypothesized segmentation. For this synthetic dataset, we will use a similar setup as before, with the dataset consisting of 1,000 reference segmentations, with an average of 50 segments per segmentation with a standard deviation of 20. Segments have an average size of 25 elements with a standard deviation of 10. However, instead of having only non-overlapping segments in the reference segmentation, we let the segments random overlap between zero and five elements to have a more realistic scenario, in which both the reference- and hypothesized segmentations have overlapping segments.

For the experiment, we start off with the hypothesized segmentation equal to the reference segmentation, but we keep increasing the overlap between the segments, up till 25, since this is the average segment size in the reference segmentations.

Figure 7.9 shows the results of the experiment with controlling the overlap in the hypothesized segmentation. The behavior of the modified metric is very similar to that of the original metric when the severity of errors is increased, showing a relatively gradual decrease in scores as the (excessive) overlap in the hypothesized segmentation increases, reaching nearly zero at the end of the graph. Although this initial experiments shows that the modified metric seems to behave appropriately, of course more experiments on real data have to be carried out to determine the efficacy of this modified version of $wF1$.

7.6 Conclusion

In this chapter, we have surveyed three groups of *extrinsic* evaluation metrics for the task of text segmentation, where reference- and hypothesized segmentations are

directly compared. Using the criteria of **meaningfulness**, **scrutiny**, **helpfulness**, and **distinctiveness**, we compared *segment-level*, *element-level*, and *sliding window* based metrics, using a combination of theoretical arguments and experiments on synthetic and real-world datasets. Although the element-level metrics (which include the well-known precision and recall on elements) are highly scrutinized and used often throughout the literature, the underlying assumptions of these metrics are flawed. Their inability to allow for small errors in segmentation and their inability to distinguish between segmentations of different qualities accurately mean that they are of little practical use when developing text segmentation systems. The methods based on a sliding window, which were developed specifically for the evaluation of text segmentation tasks, are better able to handle small errors in segmentation and to distinguish between segmentations of different qualities, due to the fact that they compare *windows* of segmentations, rather than just elements. However, the complicated formulations and the need for a parameter that controls window size mean that performing comparisons across different methods and papers is complicated. The group of metrics that best satisfies the four criteria posed in this paper is the *segment-level* group of metrics, measuring the quality of a predicted segmentation by directly comparing the segments in both the reference and hypothesized segmentations. Their direct comparisons between segments mean that the metrics can handle small errors in segmentations without the need for a hyperparameter, and that they are better in distinguishing between segmentations of different qualities. The overlap weighted metrics are particularly suited for this task, with separate scores for precision, recall, and F1 allowing for a detailed analysis of model errors, and the definition of segmentation alignment providing a simple and natural formulation of the task. Another positive aspect of the overlap weighted segment-level metrics was the ease with which they could be adapted to overlapping segmentations.

8

Conclusions

At the start of this thesis, we set out with the aim of improving public access to Dutch FOIA documents, and increasing the FAIRness of such documents at scale. As alluded to in the introduction, the current publication practices of government organizations are often not in line with the FAIR guidelines, and there is little coordination between different government agencies. In an attempt to improve the quality of these documents at scale, this thesis poses the following main research question.

Main Research Question *Can we develop enabling technology to improve FOIA document quality at scale, and how do we evaluate the quality of these extreme document clustering and segmentation tasks?*

Open government data concerns releasing recent government records from the current, state-of-the-art internal document and records management systems. This, however, is no guarantee that released open government information is in a form that satisfies common standards on *Findability*, *Accessibility*, *Interoperability*, and *Re-usability* (FAIRness) of digital assets. As a result, this thesis is focused on the post-hoc correction of Dutch FOIA documents, where we formulate two key objectives.

Key Objective 1 *To develop effective document processing technology for FOIA documents.*

Specifically, we focused on two key tasks that benefit downstream page segmentation and OCR output: page stream segmentation of complex dossiers into individual documents and redacted text detection. Both the page stream segmentation and redacted text detection tasks can be seen as instances of document segmentation and clustering tasks, and as such, evaluating methods developed for these tasks requires specific metrics to fit the task. This leads us to our second key objective.

Key Objective 2 *To Evaluate Extreme Document Segmentation and Clustering Tasks.*

In this final chapter, we will revisit the two key objectives we posed at the start of the thesis, and briefly summarize the results from each of the chapters, and where this leaves us on the main research question of the thesis. We finish with some directions for future work.

8.1 Main Findings

8.1.1 Document Processing Technology (for FOIA Documents)

In Part I of the thesis, we focused on the development of automated technology for the tasks of page stream segmentation and redacted text detection, and evaluated several techniques for their application in the domain of FOIA documents in Chapters 2 and 3. We then applied these techniques to a large dataset of Dutch FOIA documents in Chapter 4, and used these to enhance the quality of these documents at scale.

In Chapter 2 we set out to investigate different types of approaches for the task of page stream segmentation, asking the following research question:

RQ1 What is the efficacy of methods from Machine Learning for the task of Page Stream Segmentation?

We answered this question by creating a benchmark of two large public datasets of Dutch FOIA documents, after which we evaluated models from four approach types using both page- and document-level metrics. To mimic real-world usage of systems, we also conducted experiments where models were tested on data from a different source than the data they were trained on, to test their robustness to out-of-distribution data. Our experiments showed that neural networks trained using a binary page-classification scheme obtained superior performance compared to the other approach types in both the in-distribution and out-of-distribution scenarios. For the in-distribution scenario, a model that combined the textual and visual page representations proved most successful, whereas in the case of out-of-distribution data, an approach based solely on textual information proved most effective.

In Chapter 3, we continued the investigation into Machine Learning techniques for document processing tasks, tackling the task of redacted text detection. Due to the variety of possible redaction types, automatic methods that are robust to a wide variety of different types are required. Apart from a rule-based method developed by Bland et al. [15], little work has been done on developing such methods in the literature. To address this, we posed the following research question.

RQ2 What is the efficacy of neural image segmentation methods for the large-scale detection of redacted text?

We answered this research question by creating an annotated dataset of redactions and comparing two neural image segmentation models to an existing rule-based detection system. The dataset included the addition of pages without redactions, to mimic a scenario where a model is directly ran on all the documents in a collection. We found that both neural segmentation methods significantly outperform the rule-based method, while only requiring a limited amount of training data to achieve weighted F1 (i.e., Panoptic Quality) scores in excess of .93, thus showing these models can effectively detect redactions in documents. Additionally, both models produce significantly fewer false positives when tested on pages without annotations.

To investigate the feasibility of constructing a large-scale collection of FOIA documents, and to gain insights into the intricacies of the process, we constructed the Woogle

dataset in Chapter 4. Apart from providing an opportunity to test our developed methods on a large scale, it also provides us with insights into the feasibility of constructing the dataset with limited resources. In this chapter, we thus answered the following question:

RQ3 What lessons can be learned from a Living Lab of FOIA documents?

Through the construction of the Woogole dataset, we have learned valuable lessons about the process of collecting and publishing FOIA documents in the Dutch FOIA landscape. First of all, by creating a standardized set of metadata attributes and having processes to automatically extract these attributes from documents, it is possible to harmonize data from different suppliers. Secondly, the usage of off-the-shelf techniques for the processing of documents (think of PSS and the detection of redacted text) and the enhancement of text data proved good enough to publish the documents in such a way that they were useful for use in a search engine. The creation of the living lab showed that, although complicated, the large-scale construction of a collection of FOIA documents is possible.

8.1.2 Evaluation of Extreme Document Segmentation and Clustering

In Part II we explored several evaluation metrics suitable for the extreme clustering tasks described in Part I, with the discussion of alterations to the BCubed and Panoptic Quality metrics in Chapters 5 and 6, and an extensive comparison of three metric paradigms for text segmentation tasks in Chapter 7.

Chapter 5 is concerned with the BCubed evaluation metric, and aims to address some of its shortcomings, such as its inability to give a score of zero to any hypothesized segmentation.

To this extent, we posed the following question:

RQ4 Can the BCubed metric be repaired in such a way that its shortcomings are addressed while still maintaining its desirable theoretical properties?

We answered this research question in the affirmative by proposing ELM, and proving that this proposed alteration of BCubed maintains the desirable theoretical constraints of the original formulation, while also addressing its shortcomings. Due to its altered calculation, ELM allows a zero score to be obtained when, for every element in a segmentation, the overlap between the reference and hypothesized segments containing the element contains no other element.

In addition, our empirical evaluation of both metrics also showed that, when used to rank the outputs of different systems against a gold standard, ELM can produce rankings different to that of BCubed, showing that ELM is not merely a more conservative version of the BCubed metric.

Chapter 6 was concerned with the Panoptic Quality metric, focusing specifically on the matching criterion used for aligning reference and hypothesized objects when scoring a hypothesized segmentation. In the original formulation of the Panoptic Quality metric, an alignment between the reference and hypothesized segmentations is created

by matching segments that have an Intersection-over-Union score of larger than .5, ensuring a one-on-one mapping. This alignment allows for the calculation of confusion-based metrics. Although this choice of threshold seems natural, there might be other matching criteria that are more general, and still maintain this one-on-one mapping. As such, we pose the following research question:

RQ5 Is there an objective mathematical criterion for defining a matching function that ensures a one-on-one mapping between two sets of (non-overlapping) clusters?

We answer this research question in the affirmative by showing that there is in fact a most general matching criterion that maintains a one-on-one mapping over its inputs, which can be achieved by requiring the number of overlapping elements to be bigger than both the number of missed and the number of spurious elements. Apart from this, our empirical evaluation of both matching conditions shows that the more general matching rule yields more true positives when compared to the original formulation, when measured on real-world datasets. Although the difference is small, these additional true positives can be meaningful in cases where False Negatives are very costly.

As we have seen in Part I of the thesis, there are different tasks that can be formulated as a segmentation problem, and there exist many different metrics in the literature, depending on the specific task: In addition to the metrics discussed in Chapters 5 and 6, metrics such as element-based accuracy, and metrics based on sliding windows. As was the case for the task of PSS in Chapter 2, many different metrics exist, and a clear comparison of these metrics on both a theoretical and practical level was missing. As such, we posed the following question in Chapter 7:

RQ6 What is the most appropriate type of metric for the task of text segmentation?

We answer this research question by providing an extensive theoretical comparison of three groups of evaluation metrics, comparing them based on a set of qualities a ‘good’ evaluation metric should have. Apart from the theoretical comparison, we also perform a set of experiments, comparing metrics on both synthetic datasets, as well as a real-world dataset. The experimental results show that the segment-level metrics, and in particular the PQ metric, have the most desirable behavior in both cases, as they most closely align with the criteria for good evaluation metrics, without the need for specific hyperparameters.

Summary

This thesis makes important progress toward the main research problem of the thesis: *Can we develop enabling technology to improve FOIA document quality at scale, and how do we evaluate the quality of these extreme document clustering and segmentation tasks?* We divided this research problem into two clear objectives corresponding to Part I and Part II of this thesis.

Our findings of Part I on Key Objective 1 *to develop effective document processing technology for FOIA documents*, focused on creating enabling technologies for the publication of FOIA documents at scale. Our specific contributions focused on two elementary tasks that are necessary (but not sufficient) key components for effective

FOIA document processing technology. First, our contributions to page stream segmentation are key for identifying individual documents released in long PDF streams. Such segmentation is a necessary precondition for providing access to the released documents individually. Second, our contributions to the redacted text detection significantly improve the optical character recognition (OCR) output. Such OCR output is a necessary precondition for indexing and searching through the content of the released documents. Third, our contributions to publicly releasing a large set of FOIA requests as open data is an essential step toward enabling and accelerating research on FOIA search by the research community. While this thesis makes great progress on the aim of developing effective document processing technology for FOIA documents, our components may be further improved, and other components are needed to further enhance the access to FOIA documents at scale.

Our findings of Part II on Key Objective 2 *to evaluate extreme document segmentation and clustering tasks*, focused on effective evaluation of FOIA document processing technology. We have shown that by using techniques from Machine Learning, evaluated using the appropriate evaluation metrics, we can develop solutions for document processing tasks (in our case for Page Stream Segmentation and redacted text detection). Additionally, we showed that these techniques can be applied at scale, by the creation of the Woogles dataset, and that such a collection is a valuable resource for future research. Thus the results presented in this thesis show that with the proper methods and evaluation techniques, the post-hoc repairing of FOIA documents at-scale is feasible, and can contribute to closer adherence to the FAIR principles for these documents.

8.2 Future Work

This is the first thesis investigating how technology can improve public access to open government documents, and it provides only the first steps towards solving this formidable problem. We view this thesis as opening up this new research field, and hope that the data, tools, and evaluation measures will be instrumental in promoting further research on this challenging problem with the potential for great scientific, social and societal impact.

The research in this thesis can be extended in various ways in future research, for which we now give a few directions, grouped by the two parts of the thesis. For the objective *to develop effective document processing technology for FOIA documents*, future research includes the following.

First, a natural aspect of document quality is the text quality within these documents. Since the methods in which documents are produced vary greatly, there can be significant differences in the quality of the resulting text, for example, from poor quality OCR. Our preliminary work on detecting and repairing this has been done in van Heusden et al. [108], but further research is needed in developing a robust and efficient method for improving the overall text quality of FOIA documents.

Second, there is a range of other document processing tasks we have not covered in this thesis. In particular, advanced information extraction was only explored in part in Chapter 4 for metadata, but both classic and modern information extraction models have proven effective to extract, standardize and summarize key information from complex

composite dossiers of documents, even far beyond traditional metadata that is available in the original internal document or record management systems.

Third, another aspect of the digitization process that we have not covered in this thesis is developing enabling technology that can aid in the creation of these FOIA documents. Consider, for example, a system that can detect sensitive information in a document and suggest redactions. Although these technologies are not directly involved in improving the FAIRness of documents, they do aid in making the publication process more efficient, which in turn can lead to better public access to these FOIA documents.

For the objective *to evaluate extreme document segmentation and clustering tasks*, future research includes the following.

First, setting up effective evaluation measures that reflect the value of documents remains key for further developing any FOIA document processing technology. This involves both more conceptual and theoretical grounding of the measures, for example, addressing some of the remaining problems identified in Chapter 7.

Second, while scalable and reusable automatic evaluation measures are essential for technological developments, these also look at one particular aspect, typically abstracted away from the underlying use case or application. It is also important to validate the outcomes in practical use cases. One approach is by implementing prototype systems as we pioneered in Chapter 4. Another approach is to validate the findings in user-centered evaluation and user studies in the wild.

Third, evaluation typically requires large-scale labeled data, which is not always available, or sometimes not publicly shareable due to privacy or other concerns. One attractive alternative that has emerged in recent years is to use large-scale pseudo-annotated data, annotated not by human editorial judges but by using large language models with specific and detailed prompts to annotate data at scale. While this risks breaking the required independence of evaluation data, this allows for scaling up training data, and fast prototyping and developing on new data, tasks, and domains. Such LLM-Eval approaches can be a pragmatic way forward, and can be attractive and effective when combined with the user-centered validation mentioned in the previous point.

More generally, as previously mentioned, the current way documents are created and published often leads to the need for post-hoc ‘repairs’ to make them meet the FAIR data standards. In an ideal scenario, rather than resolving issues after publication, we could address these problems at the source. This requires creating standardized procedures for the publication of documents and their metadata, and having quality control mechanisms in place to guarantee high-quality data, interoperable between different suppliers. However, the sheer number of suppliers involved means that this requires complex coordination and clear best practices implemented across all the systems and practices of publishing open government data across all levels of government, rather than a simple short- or midterm solution. We can only hope that the best practices identified in the thesis will help promote the quality and (re)usability of released open government data in the future.

Bibliography

- [1] A. Abbas and P. Swoboda. Combinatorial optimization for panoptic segmentation: A fully differentiable approach. In *Advances in Neural Information Processing Systems*, volume 34, pages 15635–15649. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/83a368f54768f506b833130584455df4-Paper.pdf. (Cited on page 97.)
- [2] G. Agam, S. Argamon, O. Frieder, D. Grossman, and D. Lewis. The complex document image processing (CDIP) test collection project. *Illinois Institute of Technology*, 2006. doi: 10.18434/mds2-2531. URL <https://doi.org/10.18434/mds2-2531>. (Cited on page 17.)
- [3] E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval*, 12(4):461–486, 2009. doi: 10.1007/s10791-008-9066-8. URL <https://doi.org/10.1007/s10791-008-9066-8>. (Cited on pages 5, 67, 68, 71, 75, 77, 78, 79, 80, 81, 101, 104, 105, 121, 124, 143, and 145.)
- [4] J. Artiles, A. Borthwick, J. Gonzalo, S. Sekine, and E. Amigó. WePS-3 evaluation campaign: Overview of the web people search clustering and attribute extraction tasks. In *CLEF (Notebook Papers/LABs/Workshops)*, volume 1176 of *CEUR Workshop Proceedings*, 2010. ISBN 978-88-904810-0-0. URL <https://api.semanticscholar.org/CorpusID:17136566>. (Cited on page 80.)
- [5] A. Bagga and B. Baldwin. 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*, ACL '98/COLING '98, page 79–85, USA, 1998. Association for Computational Linguistics. doi: 10.3115/980845.980859. URL <https://doi.org/10.3115/980845.980859>. (Cited on pages 5, 67, 68, 75, 81, 102, and 104.)
- [6] F. Bakker, R. van Heusden, and M. Marx. Timeline extraction from decision letters using chatGPT. In *Proceedings of the 7th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2024)*, pages 24–31, St. Julians, Malta, Mar. 2024. Association for Computational Linguistics. URL <https://aclanthology.org/2024.case-1.3>.
- [7] A.-L. Barabási and M. Pósfai. *Network Science*. Cambridge University Press, Cambridge, 2016. ISBN 9781107076266 1107076269. URL <http://barabasi.com/networksciencebook/>. (Cited on page 82.)
- [8] J. Barrow, R. Jain, V. Morariu, V. Manjunatha, D. Oard, and P. Resnik. A joint model for document segmentation and segment labeling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 313–322, New York, USA, 2020. Association for Computational Linguistics. URL <https://aclanthology.org/2020.acl-main.29>. (Cited on page 17.)
- [9] D. Beeferman, A. Berger, and J. Lafferty. Statistical models for text segmentation. *Mach. Learn.*, 34(1–3):177–210, Feb. 1999. ISSN 0885-6125. doi: 10.1023/A:1007506220214. URL <https://doi.org/10.1023/A:1007506220214>. (Cited on pages 19, 20, 71, 103, 108, 113, and 114.)
- [10] S.-M.-R. Beheshti, B. Benatallah, S. Venugopal, S. H. Ryu, H. R. Motahari-Nezhad, and W. Wang. A systematic review and comparative analysis of cross-document coreference resolution methods and tools. *Computing*, 99(4):313–349, Apr. 2017. ISSN 0010-485X. doi: 10.1007/s00607-016-0490-0. URL <https://doi.org/10.1007/s00607-016-0490-0>. (Cited on page 80.)
- [11] M. Bernhard, R. Amoroso, Y. Kindermann, L. Baraldi, R. Cucchiara, V. Tresp, and M. Schubert. What’s outside the intersection? fine-grained error analysis for semantic segmentation beyond iou. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 957–966, 2024. doi: 10.1109/WACV57701.2024.00101. URL <https://doi.org/10.1109/WACV57701.2024.00101>. (Cited on page 94.)
- [12] E. Bier, R. Chow, P. Gollé, T. H. King, and J. Staddon. The rules of redaction: Identify, protect, review (and repeat). *IEEE Security & Privacy*, 7(6):46–53, 2009. doi: 10.1109/MSP.2009.183. URL <https://doi.org/10.1109/MSP.2009.183>. (Cited on page 32.)
- [13] S. Biswas, P. Riba, J. Lladós, and U. Pal. Beyond document object detection: Instance-level segmentation of complex layouts. *International Journal on Document Analysis and Recognition*, 24(3):269–281, 2021. doi: 10.1007/s10032-021-00380-6. URL <https://doi.org/10.1007/s10032-021-00380-6>. (Cited on pages 35 and 37.)
- [14] S. Biswas, A. Banerjee, J. Lladós, and U. Pal. Docsegr: An instance-level end-to-end document image segmentation transformer. *arXiv preprint arXiv:2201.11438*, 2022. doi: 10.48550/arXiv.2201.11438. URL <https://doi.org/10.48550/arXiv.2201.11438>. (Cited on pages 35 and 38.)
- [15] M. Bland, A. Iyer, and K. Levchenko. Story beyond the eye: Glyph positions break PDF text redaction. *Proceedings on Privacy Enhancing Technologies*, 2023(3):43–61, 2023. URL <https://doi.org/10.56553/popets-2023-0069>. (Cited on pages 32, 33, 34, 38, 47, and 128.)

8. Bibliography

- [16] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee. Yolact: Real-time instance segmentation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9156–9165, 2019. doi: 10.1109/ICCV.2019.00925. URL <https://doi.org/10.1109/ICCV.2019.00925>. (Cited on page 35.)
- [17] W. M. Bramer, G. B. De Jonge, M. L. Rethlefsen, F. Mast, and J. Kleijnen. A systematic approach to searching: an efficient and complete method to develop literature searches. *Journal of the Medical Library Association: JMLA*, 106(4):531, 2018. doi: 10.5195/jmla.2018.283. URL <https://doi.org/10.5195/jmla.2018.283>. (Cited on page 52.)
- [18] F. Braz, N. Silva, and J. A. Lima. Leveraging effectiveness and efficiency in page stream deep segmentation. *Engineering Applications of Artificial Intelligence*, 105:104394, 10 2021. doi: 10.1016/j.engappai.2021.104394. URL <https://doi.org/10.1016/j.engappai.2021.104394>. (Cited on pages 16, 17, 18, 21, and 56.)
- [19] L. Busch, R. van Heusden, and M. Marx. Using deep-learned vector representations for page stream segmentation by agglomerative clustering. *Algorithms*, 16(5):259, 2023. doi: 10.3390/a16050259. URL <https://doi.org/10.3390/a16050259>.
- [20] A. Chaney, H. Wallach, M. Connelly, and D. Blei. Detecting and characterizing events. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1142–1152, 2016. doi: 10.18653/v1/D16-1122. URL <https://doi.org/10.18653/v1/D16-1122>. (Cited on page 52.)
- [21] J. Chantal, S. Hercberg, WHO, et al. Development of a new front-of-pack nutrition label in france: the five-colour nutri-score. *Public health panorama*, 3(04):712–725, 2017. URL <https://iris.who.int/handle/10665/325207>. (Cited on page 55.)
- [22] H. Chen, K. Sun, Z. Tian, C. Shen, Y. Huang, and Y. Yan. BlendMask: Top-down meets bottom-up for instance segmentation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8570–8578, 2020. doi: 10.1109/CVPR42600.2020.00860. URL <https://doi.org/10.1109/CVPR42600.2020.00860>. (Cited on pages 34 and 35.)
- [23] L. Chen, Y. Wu, J. Stegmaier, and D. Merhof. SortedAP: Rethinking evaluation metrics for instance segmentation. In *2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 3925–3931, 2023. doi: 10.1109/ICCVW60793.2023.00424. URL <https://doi.org/10.1109/ICCVW60793.2023.00424>. (Cited on page 86.)
- [24] T. Chen and C. Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16*, page 785–794, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939785. URL <https://doi.org/10.1145/2939672.2939785>. (Cited on pages 20 and 116.)
- [25] B. Cheng, R. Girshick, P. Dollár, A. C. Berg, and A. Kirillov. Boundary iou: Improving object-centric image segmentation evaluation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15329–15337, 2021. doi: 10.1109/CVPR46437.2021.01508. URL <https://doi.org/10.1109/CVPR46437.2021.01508>. (Cited on page 96.)
- [26] B. Cheng, A. G. Schwing, and A. Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *Proceedings of the 35th International Conference on Neural Information Processing Systems, NIPS '21*, Red Hook, NY, USA, 2021. Curran Associates Inc. ISBN 9781713845393. (Cited on pages 37, 38, and 95.)
- [27] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar. Masked-attention mask transformer for universal image segmentation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1280–1289, 2022. doi: 10.1109/CVPR52688.2022.00135. URL <https://doi.org/10.1109/CVPR52688.2022.00135>. (Cited on pages 32 and 37.)
- [28] F. Y. Y. Choi. Advances in domain independent linear rext segmentation. In *1st Meeting of the North American Chapter of the Association for Computational Linguistics*, 2000. URL <https://aclanthology.org/A00-2004>. (Cited on page 19.)
- [29] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The Cityscapes dataset for semantic urban scene understanding. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, 2016. doi: 10.1109/CVPR.2016.350. URL <https://doi.org/10.1109/CVPR.2016.350>. (Cited on pages 92 and 95.)
- [30] H. Daher and A. Belaïd. Document flow segmentation for business applications. In *Document Recognition and Retrieval XXI*, volume 9021, page 90210. International Society for Optics and Photonics, SPIE, 2014. doi: 10.1117/12.2043141. URL <https://doi.org/10.1117/12.2043141>. (Cited on page 17.)

-
- [31] F. J. Damerau. A technique for computer detection and correction of spelling errors. *Communications of the ACM*, 7(3):171–176, 1964. doi: 10.1145/363958.363994. URL <https://doi.org/10.1145/363958.363994>. (Cited on page 103.)
- [32] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848. URL <https://doi.org/10.1109/CVPR.2009.5206848>. (Cited on page 21.)
- [33] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186, Minneapolis, MN, USA, 2019. Association for Computational Linguistics. URL <https://doi.org/10.18653/v1/n19-1423>. (Cited on pages 16 and 74.)
- [34] G. O. Diaz and J. Ouyang. An alignment-based approach to text segmentation similarity scoring. In *Proceedings of the 26th Conference on Computational Natural Language Learning*, pages 374–383, Abu Dhabi, United Arab Emirates, 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.conll-1.26. URL <https://doi.org/10.18653/v1/2022.conll-1.26>. (Cited on pages 102, 104, 106, and 112.)
- [35] A. Dutta and A. Zisserman. The VIA annotation software for images, audio and video. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM ’19, New York, NY, USA, 2019. ACM. ISBN 978-1-4503-6889-6/19/10. doi: 10.1145/3343031.3350535. URL <https://doi.org/10.1145/3343031.3350535>. (Cited on page 36.)
- [36] T. Erjavec, M. Ogrodniczuk, P. Osenova, N. Ljubesic, K. Simov, A. Pancur, M. Rudolf, M. Kopp, S. Barkarson, S. Steingrímsson, Ç. Çöltekin, J. de Does, K. Depuydt, T. Agnoloni, G. Venturi, M. C. Pérez, L. D. de Macedo, C. Navarretta, G. Luxardo, M. Coole, P. Rayson, V. Morkevicius, T. Krilavicius, R. Dargis, O. Ring, R. van Heusden, M. Marx, and D. Fiser. The parlamin corpora of parliamentary proceedings. *Language Resources and Evaluation*, 57(1):415–448, 2023. doi: 10.1007/s10579-021-09574-0. URL <https://doi.org/10.1007/s10579-021-09574-0>.
- [37] T. Erjavec, M. Kopp, N. Ljubešić, T. Kuzman, P. Rayson, P. Osenova, M. Ogrodniczuk, Ç. Çöltekin, D. Koržinek, K. Meden, J. Skubic, P. Rupnik, T. Agnoloni, J. Aires, S. Barkarson, R. Bartolini, N. Bel, M. C. Pérez, R. Dargis, S. Diwersy, M. Gavrilidou, R. van Heusden, M. Iruskieta, N. Kahusk, A. Kryvenko, N. Ligeti-Nagy, C. Magariños, M. Mölder, C. Navarretta, K. Simov, L. M. Tunlgand, J. Tuominen, J. Vidler, A. I. Vladu, T. Wissik, V. Yrjänäinen, and D. Fišer. Parlamin II: advancing comparable parliamentary corpora across europe. *Language Resources and Evaluation*, pages 1–32, 2024. ISSN 1574-0218. doi: 10.1007/s10579-024-09798-w. URL <https://doi.org/10.1007/s10579-024-09798-w>.
- [38] A. A. Ferreira, M. A. Gonçalves, and A. H. Laender. A brief survey of automatic methods for author name disambiguation. *SIGMOD Record*, 41(2):15–26, Aug. 2012. ISSN 0163-5808. doi: 10.1145/2350036.2350040. URL <https://doi.org/10.1145/2350036.2350040>. (Cited on page 80.)
- [39] W. K. Fong, R. Mohan, J. V. Hurtado, L. Zhou, H. Caesar, O. Beijbom, and A. Valada. Panoptic nusenes: A large-scale benchmark for lidar panoptic segmentation and tracking. *IEEE Robotics and Automation Letters*, 7(2):3795–3802, 2022. doi: 10.1109/LRA.2022.3148457. URL <https://doi.org/10.1109/LRA.2022.3148457>. (Cited on page 96.)
- [40] A. Foucart, O. Debeir, and C. Decaestecker. Panoptic quality should be avoided as a metric for assessing cell nuclei segmentation and classification in digital pathology. *Scientific Reports*, 13(1):8614, 2023. doi: 10.1038/s41598-023-35605-7. URL <https://doi.org/10.1038/s41598-023-35605-7>. (Cited on pages 95 and 97.)
- [41] C. Fournier. valuating text segmentation using boundary edit distance. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 1702–1712, Sofia, Bulgaria, 2013. Association for Computational Linguistics. URL <https://aclanthology.org/P13-1167/>. (Cited on pages 103, 104, 107, and 108.)
- [42] C. Fournier and D. Inkpen. Segmentation similarity and agreement. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 152–161. Association for Computational Linguistics, 2012. (Cited on pages 103, 104, 107, and 114.)
- [43] M. Georgescu, A. Clark, and S. Armstrong. Word distributions for thematic segmentation in a Support Vector Machine approach. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pages 101–108, New York City, June 2006. Association for Computational Linguistics. URL <https://aclanthology.org/W06-2914/>. (Cited on page 103.)

- [44] I. Ghinassi, L. Wang, C. Newell, and M. Purver. Lessons learnt from linear text segmentation: a fair comparison of architectural and sentence encoding strategies for successful segmentation. In *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, pages 408–418, Varna, Bulgaria, Sept. 2023. INCOMA Ltd., Shoumen, Bulgaria. URL <https://aclanthology.org/2023.ranlp-1.46/>. (Cited on page 103.)
- [45] R. Girshick. Fast R-CNN. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015. doi: 10.1109/ICCV.2015.169. URL <https://doi.org/10.1109/ICCV.2015.169>. (Cited on pages 37 and 95.)
- [46] A. Guha, A. Alahmadi, D. Samanta, M. Z. Khan, and A. H. Alahmadi. A multi-modal approach to digital document stream segmentation for title insurance domain. *IEEE Access*, 10:11341–11353, 2022. doi: 10.1109/ACCESS.2022.3144185. URL <https://doi.org/10.1109/ACCESS.2022.3144185>. (Cited on pages 16, 17, 21, 56, and 75.)
- [47] A. Gupta, P. Dollár, and R. Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5351–5359, 2019. doi: 10.1109/CVPR.2019.00550. URL <https://doi.org/10.1109/CVPR.2019.00550>. (Cited on page 92.)
- [48] I. Habernal and I. Gurevych. Argumentation mining in user-generated web discourse. *Computational Linguistics*, 43(1):125–179, Apr. 2017. doi: 10.1162/COLLa.00276. URL <https://aclanthology.org/J17-1004/>. (Cited on page 103.)
- [49] A. Hamdi, M. Coustaty, A. Joseph, V. P. d’Andecy, A. Doucet, and J.-M. Ogier. Feature selection for document flow segmentation. In *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, pages 245–250, 2018. doi: 10.1109/DAS.2018.66. URL <https://doi.org/10.1109/DAS.2018.66>. (Cited on page 17.)
- [50] R. W. Hamming. Error detecting and error correcting codes. *The Bell System Technical Journal*, 29(2):147–160, 1950. doi: 10.1002/j.1538-7305.1950.tb00463.x. URL <https://doi.org/10.1002/j.1538-7305.1950.tb00463.x>. (Cited on pages 103 and 107.)
- [51] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90. URL <https://doi.org/10.1109/CVPR.2016.90>. (Cited on page 34.)
- [52] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017. doi: 10.1109/ICCV.2017.322. URL <https://doi.org/10.1109/ICCV.2017.322>. (Cited on pages 32, 35, 37, 92, and 95.)
- [53] H. Hernault, D. Bollegala, and M. Ishizuka. A sequential model for discourse segmentation. In *Computational Linguistics and Intelligent Text Processing*, pages 315–326, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-12116-6. doi: 10.1007/978-3-642-12116-6_26. URL https://doi.org/10.1007/978-3-642-12116-6_26. (Cited on page 18.)
- [54] S. Hill, Z. Zhou, L. Saul, and H. Shacham. On the (in) effectiveness of mosaicing and blurring as tools for document redaction. *Proceedings on Privacy Enhancing Technologies*, 2016. doi: 10.1515/popets-2016-0047. URL <https://doi.org/10.1515/popets-2016-0047>. (Cited on page 32.)
- [55] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computing*, 9(8):1735–1780, nov 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL <https://doi.org/10.1162/neco.1997.9.8.1735>. (Cited on page 16.)
- [56] Y. Huang, T. Lv, L. Cui, Y. Lu, and F. Wei. LayoutLMv3: Pre-training for document AI with unified text and image masking. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM ’22, page 4083–4091, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392037. doi: 10.1145/3503161.3548112. URL <https://doi.org/10.1145/3503161.3548112>. (Cited on page 35.)
- [57] H. Ji, R. Grishman, H. T. Dang, K. Griffitt, and J. Ellis. Overview of the TAC 2010 knowledge base population track. In *Proceedings of the Third Text Analysis Conference (2010)*, volume 3, pages 3–3, 2010. URL <https://api.semanticscholar.org/CorpusID:854997>. (Cited on page 80.)
- [58] H. Ji, J. Nothman, and B. e. a. Hachey. Overview of TAC-KBP2014 entity discovery and linking tasks. In *Proceedings of the Text Analysis Conference (2014)*, pages 1333–1339, 2014. URL <https://api.semanticscholar.org/CorpusID:17473892>. (Cited on page 80.)
- [59] D. Kim, S. Woo, J.-Y. Lee, and I. S. Kweon. Video panoptic segmentation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9856–9865, 2020. doi: 10.1109/CVPR42600.2020.00988. URL <https://doi.org/10.1109/CVPR42600.2020.00988>. (Cited on page 96.)
- [60] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár. Panoptic segmentation. In *2019 IEEE/CVF*

- Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9396–9405, 2019. doi: 10.1109/CVPR.2019.00963. URL <https://doi.org/10.1109/CVPR.2019.00963>. (Cited on pages 5, 8, 22, 33, 39, 40, 83, 84, 87, 100, 101, 102, and 104.)
- [61] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick. Segment anything. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3992–4003, 2023. doi: 10.1109/ICCV51070.2023.00371. URL <https://doi.org/10.1109/ICCV51070.2023.00371>. (Cited on page 95.)
- [62] J. Kleinberg. Bursty and hierarchical structure in streams. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '02*, page 91–101, New York, NY, USA, 2002. Association for Computing Machinery. ISBN 158113567X. doi: 10.1145/775047.775061. URL <https://doi.org/10.1145/775047.775061>. (Cited on page 103.)
- [63] O. Koshorek, A. Cohen, N. Mor, M. Rotman, and J. Berant. Text segmentation as a supervised learning task. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 469–473, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. URL <https://aclanthology.org/N18-2075>. (Cited on pages 18, 21, and 116.)
- [64] L. Kuncheva. A theoretical study on six classifier fusion strategies. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):281–286, 2002. doi: 10.1109/34.982906. URL <https://doi.org/10.1109/34.982906>. (Cited on pages 22 and 27.)
- [65] S. Lamprier, T. Amghar, B. Levrat, and F. Saubion. On evaluation methodologies for text segmentation algorithms. In *Proceedings of the 19th IEEE international conference on tools with artificial intelligence*, volume 2, pages 19–26, Patras, Greece, 2007. IEEE. doi: 10.1109/ICTAI.2007.22. URL <https://doi.org/10.1109/ICTAI.2007.22>. (Cited on page 109.)
- [66] Q. Le and T. Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32, ICML'14*, page II–1188–II–1196. JMLR.org, 2014. (Cited on page 21.)
- [67] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710, Feb. 1966. URL <https://api.semanticscholar.org/CorpusID:60827152>. (Cited on pages 103 and 107.)
- [68] D. Lewis, G. Agam, S. Argamon, O. Frieder, D. Grossman, and J. Heard. 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*, page 665–666, New York, NY, USA, 2006. Association for Computing Machinery. URL <https://doi.org/10.1145/1148170.1148307>. (Cited on page 17.)
- [69] J. Li, Y. Xu, T. Lv, L. Cui, C. Zhang, and F. Wei. Dit: Self-supervised pre-training for document image transformer. In *Proceedings of the 30th ACM International Conference on Multimedia, MM '22*, page 3530–3539, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392037. doi: 10.1145/3503161.3547911. URL <https://doi.org/10.1145/3503161.3547911>. (Cited on page 35.)
- [70] X. Li, H. Ding, H. Yuan, W. Zhang, J. Pang, G. Cheng, K. Chen, Z. Liu, and C. C. Loy. Transformer-based visual segmentation: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 46(12):10138–10163, Dec. 2024. ISSN 0162-8828. doi: 10.1109/TPAMI.2024.3434373. URL <https://doi.org/10.1109/TPAMI.2024.3434373>. (Cited on page 95.)
- [71] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10602-1. doi: 10.1007/978-3-319-10602-1_48. URL https://doi.org/10.1007/978-3-319-10602-1_48. (Cited on pages 37, 40, and 92.)
- [72] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, 2017. doi: 10.1109/CVPR.2017.106. URL <https://doi.org/10.1109/CVPR.2017.106>. (Cited on page 37.)
- [73] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia. Path aggregation network for instance segmentation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8759–8768, 2018. doi: 10.1109/CVPR.2018.00913. URL <https://doi.org/10.1109/CVPR.2018.00913>. (Cited on page 95.)
- [74] Y. Liu, C. Si, K. Jin, T. Shen, and M. Hu. FCENet: An instance segmentation model for extracting figures and captions from material documents. *IEEE Access*, 9:551–564, 2021. doi: 10.1109/ACCESS.

- 2020.3046496. URL <https://doi.org/10.1109/ACCESS.2020.3046496>. (Cited on page 34.)
- [75] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9992–10002, 2021. doi: 10.1109/ICCV48922.2021.00986. URL <https://doi.org/10.1109/ICCV48922.2021.00986>. (Cited on page 38.)
- [76] D. Lopresti and A. L. Spitz. Quantifying information leakage in document redaction. In *Proceedings of the 1st ACM Workshop on Hardcopy Document Processing*, HDP '04, page 63–69, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 1581139764. doi: 10.1145/1031442.1031452. URL <https://doi.org/10.1145/1031442.1031452>. (Cited on page 32.)
- [77] M. Lukasik, B. Dadachev, K. Papineni, and G. Simões. Text segmentation by cross segment attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4707–4716, Online, Nov. 2020. Association for Computational Linguistics. URL <https://aclanthology.org/2020.emnlp-main.380>. (Cited on page 18.)
- [78] T. Meilender and A. Belaïd. Segmentation of continuous document flow by a modified backward-forward algorithm. In *Document Recognition and Retrieval XVI*, volume 7247, page 724705. International Society for Optics and Photonics, SPIE, 2009. doi: 10.1117/12.805646. URL <https://doi.org/10.1117/12.805646>. (Cited on page 17.)
- [79] F. Menczer, S. Fortunato, and C. A. Davis. *A First Course in Network Science*. Cambridge University Press, Cambridge, 2020. doi: 10.1017/9781108653947. URL <https://10.1017/9781108653947>. (Cited on page 82.)
- [80] A. Moffat. Seven numeric properties of effectiveness metrics. In *Proceedings of the 9th Asia Information Retrieval Societies Conference*, volume 9, pages 1–12, Singapore, 2013. Springer. doi: 10.1007/978-3-642-45068-6_1. URL https://doi.org/10.1007/978-3-642-45068-6_1. (Cited on pages 101 and 102.)
- [81] J. G. Moreno and G. Dias. Adapted b-cubed metrics to unbalanced datasets. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, page 911–914, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450336215. doi: 10.1145/2766462.2767836. URL <https://doi.org/10.1145/2766462.2767836>. (Cited on page 80.)
- [82] G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005. doi: 10.1038/nature03607. URL <https://doi.org/10.1038/nature03607>. (Cited on page 124.)
- [83] P. Pantel and D. Lin. Efficiently clustering documents with committees. In M. Ishizuka and A. Sattar, editors, *PRICAI 2002: Trends in Artificial Intelligence*, pages 424–433, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg. ISBN 978-3-540-45683-4. (Cited on page 59.)
- [84] L. Pevzner and M. A. Hearst. Critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, 28(1):19–36, 2002. URL <https://aclanthology.org/J02-1002>. (Cited on pages 101, 102, 103, and 109.)
- [85] L. L. Pipino, Y. W. Lee, and R. Y. Wang. Data quality assessment. *Communications of the ACM*, 45(4):211–218, Apr. 2002. ISSN 0001-0782. doi: 10.1145/505248.506010. URL <https://doi.org/10.1145/505248.506010>. (Cited on page 59.)
- [86] C. Poot and A. van Cranenburgh. A benchmark of rule-based and neural coreference resolution in Dutch novels and news. In *Proceedings of the Third Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 79–90, Barcelona, Spain (online), Dec. 2020. Association for Computational Linguistics. URL <https://aclanthology.org/2020.crac-1.9/>. (Cited on page 80.)
- [87] A. Rahman and V. Ng. Supervised models for coreference resolution. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 968–977, Singapore, Aug. 2009. Association for Computational Linguistics. URL <https://aclanthology.org/D09-1101/>. (Cited on page 80.)
- [88] D. Ramachandram and G. W. Taylor. Deep multimodal learning: A survey on recent advances and trends. *IEEE Signal Processing Magazine*, 34(6):96–108, 2017. doi: 10.1109/MSP.2017.2738401. URL <https://doi.org/10.1109/MSP.2017.2738401>. (Cited on page 22.)
- [89] J. C. Reynar. An automatic method of finding topic boundaries. In *32nd Annual Meeting of the Association for Computational Linguistics*, pages 331–333, Las Cruces, New Mexico, USA, June 1994. Association for Computational Linguistics. URL <https://aclanthology.org/P94-1050>. (Cited on page 19.)
- [90] H. Rosales-Méndez and Y. Ramírez-Cruz. CICE-BCubed: A new evaluation measure for overlapping

-
- clustering algorithms. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pages 157–164. Springer, 2013. ISBN 978-3-642-41822-8. doi: 10.1007/978-3-642-41822-8_20. URL https://doi.org/10.1007/978-3-642-41822-8_20. (Cited on pages 81 and 124.)
- [91] J. Sauvola, T. Seppanen, S. Haapakoski, and M. Pietikainen. Adaptive document binarization. In *Proceedings of the Fourth International Conference on Document Analysis and Recognition*, volume 1, pages 147–152, 1997. doi: 10.1109/ICDAR.1997.619831. URL <https://doi.org/10.1109/ICDAR.1997.619831>. (Cited on page 19.)
- [92] M. Scaiano and D. Inkpen. Getting more from segmentation evaluation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 362–366, Montréal, Canada, June 2012. Association for Computational Linguistics. URL <https://aclanthology.org/N12-1038/>. (Cited on pages 103 and 109.)
- [93] A. J. Sharkey. *Combining Artificial Neural Nets: Ensemble and Modular Multi-Net Systems*. Springer-Verlag, Berlin, Heidelberg, 1st edition, 1999. ISBN 185233004X. (Cited on page 26.)
- [94] D. J. Sheffner. *The Freedom of Information Act (FOIA): A Legal Overview*. Congressional Research Service, 2020. (Cited on page 50.)
- [95] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. URL <https://api.semanticscholar.org/CorpusID:14124313>. (Cited on page 34.)
- [96] N. Stylianou and I. Vlahavas. A neural entity coreference resolution review. *Expert Systems with Applications*, 168:114466, 2021. ISSN 0957-4174. doi: <https://doi.org/10.1016/j.eswa.2020.114466>. URL <https://www.sciencedirect.com/science/article/pii/S0957417420311143>. (Cited on page 80.)
- [97] M. Tan and Q. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *36th International Conference on Machine Learning (ICML)*, pages 6105–6114. PMLR, 2019. URL <https://api.semanticscholar.org/CorpusID:167217261>. (Cited on page 17.)
- [98] A. Tversky. Features of similarity. *Psychological Review*, 84(4):327–352, 1977. doi: 10.1037/0033-295X.84.4.327. URL <https://doi.org/10.1037/0033-295X.84.4.327>. (Cited on page 89.)
- [99] UNESCO. Global report on the implementation of access to information laws. In *Intergovernmental Council of the International Programme for the Development of Communication*, 33rd, Paris, 2022, pages 1–12, 2022. (Cited on page 49.)
- [100] O. Uzuner, A. Bodnari, S. Shen, T. Forbush, J. Pestian, and B. R. South. Evaluating the state of the art in coreference resolution for electronic medical records. *Journal of the American Medical Informatics Association*, 19(5):786–791, 02 2012. ISSN 1067-5027. doi: 10.1136/amiajnl-2011-000784. URL <https://doi.org/10.1136/amiajnl-2011-000784>. (Cited on page 80.)
- [101] R. van Heusden and M. Marx. A sharper definition of alignment for panoptic quality. *Pattern Recognition Letters*, 185:87–93, 2024. doi: 10.1016/j.patrec.2024.07.005. URL <https://doi.org/10.1016/j.patrec.2024.07.005>. (Cited on page 100.)
- [102] R. van Heusden and M. Marx. Text segmentation metrics: A survey, 2025. To be submitted.
- [103] R. van Heusden, J. Kamps, and M. Marx. Bcubed revisited: Elements like me. In *ICTIR '22: The 2022 ACM SIGIR International Conference on the Theory of Information Retrieval, Madrid, Spain, July 11 - 12, 2022*, pages 127–132. ACM, 2022. doi: 10.1145/3539813.3545121. URL <https://doi.org/10.1145/3539813.3545121>.
- [104] R. van Heusden, J. Kamps, and M. Marx. Woolir: A new open page stream segmentation dataset. In *ICTIR '22: The 2022 ACM SIGIR International Conference on the Theory of Information Retrieval, Madrid, Spain, July 11 - 12, 2022*, pages 24–33. ACM, 2022. doi: 10.1145/3539813.3545150. URL <https://doi.org/10.1145/3539813.3545150>. (Cited on page 18.)
- [105] R. van Heusden, M. Marx, and J. Kamps. Entity linking in the parlamin corpus. In *Proceedings of the Workshop ParlaCLARIN III within the 13th Language Resources and Evaluation Conference*, pages 47–55, Marseille, France, June 2022. European Language Resources Association. URL <https://aclanthology.org/2022.parlaclarin-1.8/>.
- [106] R. van Heusden, A. de Ruijter, R. Majoor, and M. Marx. Detection of redacted text in legal documents. In *International Conference on Theory and Practice of Digital Libraries*, pages 310–316. Springer, 2023. doi: 10.1007/978-3-031-43849-3_28. URL https://doi.org/10.1007/978-3-031-43849-3_28. (Cited on pages 33, 34, 35, 38, 57, and 58.)
- [107] R. van Heusden, J. Kamps, and M. Marx. Neural coreference resolution for dutch parliamentary documents with the DutchParliament dataset. *Data*, 8(2):34, 2023. doi: 10.3390/data8020034. URL <https://doi.org/10.3390/data8020034>.

- [108] R. van Heusden, H. Ling, L. Nelissen, and M. Marx. Making PDFs accessible for visually impaired users (and findable for everybody else). In *International Conference on Theory and Practice of Digital Libraries*, pages 239–245. Springer, 2023. doi: 10.1007/978-3-031-43849-3_21. URL https://doi.org/10.1007/978-3-031-43849-3_21. (Cited on page 131.)
- [109] R. van Heusden, J. Kamps, and M. Marx. Bcubed revisited: Elements like me. *Discover Computing*, 27(1):5, 2024. doi: 10.1007/s10791-024-09436-7. URL <https://doi.org/10.1007/s10791-024-09436-7>. (Cited on pages 102, 104, and 105.)
- [110] R. van Heusden, J. Kamps, and M. Marx. OpenPSS: An open page stream segmentation benchmark. In *Linking Theory and Practice of Digital Libraries: 28th International Conference on Theory and Practice of Digital Libraries, TPDL 2024, Ljubljana, Slovenia, September 24–27, 2024, Proceedings, Part I*, pages 413–429. Springer, 2024. doi: 10.1007/978-3-031-72437-4_24. URL https://doi.org/10.1007/978-3-031-72437-4_24. (Cited on pages 56, 57, 115, and 116.)
- [111] R. van Heusden, M. Larooij, J. Kamps, and M. Marx. A collection of fair dutch freedom of information act documents (v4). *DANS Data Station Social Sciences and Humanities*, 2025. doi: 10.17026/dans-zau-e3rk. URL <https://doi.org/10.17026/dans-zau-e3rk>. (Cited on page 58.)
- [112] R. van Heusden, M. Larooij, J. Kamps, and M. Marx. A collection of fair dutch freedom of information act documents. *Scientific Data*, 12(1):795, 2025. ISSN 2052-4463. doi: 10.1038/s41597-025-05052-2. URL <https://doi.org/10.1038/s41597-025-05052-2>.
- [113] R. van Heusden, K. Meijer, and M. Marx. Redacted text detection using neural image segmentation models. *International Journal on Document Analysis and Recognition (IJDAR)*, 2025. URL <https://doi.org/10.1007/s10032-025-00513-1>.
- [114] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 6000–6010. Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964. (Cited on pages 29, 33, and 116.)
- [115] R. Verma, N. Kumar, A. Patil, N. C. Kurian, S. Rane, and A. Sethi. Multi-organ nuclei segmentation and classification challenge 2020. *IEEE Transactions on Medical Imaging*, 39(1380-1391):3413–3423, 2020. doi: 10.1109/TMI.2021.3085712. URL <https://doi.org/10.1109/TMI.2021.3085712>. (Cited on page 96.)
- [116] G. Viira and M. Marx. Enhancing access across europe for documents published according to freedom of information act: Applying woogle design and technique to estonian public information act document. *Data*, 9(11), 2024. ISSN 2306-5729. doi: 10.3390/data9110125. URL <https://www.mdpi.com/2306-5729/9/11/125>. (Cited on page 52.)
- [117] M. Vilain, J. Burger, J. Aberdeen, D. Connolly, and L. Hirschman. A model-theoretic coreference scoring scheme. In *Proceedings of the 6th Conference on Message Understanding, MUC6 '95*, page 45–52. USA, 1995. Association for Computational Linguistics. ISBN 1558604022. doi: 10.3115/1072399.1072405. URL <https://doi.org/10.3115/1072399.1072405>. (Cited on page 80.)
- [118] E. Wagner, R. Keydar, A. Pinchevski, and O. Abend. Topical segmentation of spoken narratives: A test case on holocaust survivor testimonies. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6809–6821, Abu Dhabi, United Arab Emirates, Dec. 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.457. URL <https://aclanthology.org/2022.emnlp-main.457/>. (Cited on page 103.)
- [119] Y. Wang, S. Li, and J. Yang. Toward fast and accurate neural discourse segmentation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 962–967, Brussels, Belgium, 2018. Association for Computational Linguistics. URL <https://aclanthology.org/D18-1116>. (Cited on page 18.)
- [120] J. Weyler, F. Magistri, E. Marks, Y. L. Chong, M. Sodano, G. Roggiolani, N. Chebrolu, C. Stachniss, and J. Behley. PhenoBench — A Large Dataset and Benchmarks for Semantic Image Interpretation in the Agricultural Domain. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12): 9583–9594, 2024. (Cited on page 96.)
- [121] G. Wiedemann and G. Heyer. Multi-modal page stream segmentation with convolutional neural networks. *Language Resource Evaluation*, 55(1):127–150, Mar. 2021. ISSN 1574-020X. doi: 10.1007/s10579-019-09476-2. URL <https://doi.org/10.1007/s10579-019-09476-2>. (Cited on pages 16, 17, 21, 56, 75, and 116.)
- [122] M. D. Wilkinson, M. Dumontier, I. J. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, J. Bouwman, A. J. Brookes, T. Clark, M. Crosas, I. Dillo, O. Dumon, S. Edmunds, C. T. Evelo, R. Finkers, A. Gonzalez-Beltran, A. J. G. Gray, P. Groth, C. Goble, J. S. Grethe, J. Heringa, P. A. C. 't Hoen, R. Hooft, T. Kuhn, R. Kok, J. Kok, S. J. Lusher, M. E.

-
- Martone, A. Mons, A. L. Packer, B. Persson, P. Rocca-Serra, M. Roos, R. van Schaik, S.-A. Sansone, E. Schultes, T. Sengstag, T. Slater, G. Strawn, M. A. Swertz, M. Thompson, J. van der Lei, E. van Mulligen, J. Velterop, A. Waagmeester, P. Wittenburg, K. Wolstencroft, J. Zhao, and B. Mons. The fair guiding principles for scientific data management and stewardship. *Scientific Data*, 3:160018, 2016. doi: 10.1038/sdata.2016.18. URL <https://www.nature.com/articles/sdata201618>. (Cited on pages 1 and 50.)
- [123] J. Wolswinkel. Actieve openbaarmaking van beschikkingen: Op weg naar transparante besluitvorming ‘nieuwe stijl’? *Nederlands Juristenblad*, 2024(24):1851–1857, July 2024. ISSN 0165-0483. (Cited on page 50.)
- [124] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. Detectron2. <https://github.com/facebookresearch/detectron2>, 2019. (Cited on pages 37 and 92.)
- [125] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5987–5995, 2017. doi: 10.1109/CVPR.2017.634. URL <https://doi.org/10.1109/CVPR.2017.634>. (Cited on page 37.)
- [126] Y. Xiong, R. Liao, H. Zhao, R. Hu, M. Bai, E. Yumer, and R. Urtasun. UPSNet: A unified panoptic segmentation network. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8810–8818, 2019. doi: 10.1109/CVPR.2019.00902. URL <https://doi.org/10.1109/CVPR.2019.00902>. (Cited on page 97.)
- [127] G. Zhu and D. Doermann. Automatic document logo detection. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 2, pages 864–868, 2007. doi: 10.1109/ICDAR.2007.4377038. URL <https://doi.org/10.1109/ICDAR.2007.4377038>. (Cited on page 18.)
- [128] G. Zhu, Y. Zheng, D. Doermann, and S. Jaeger. Multi-scale structural saliency for signature detection. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007. doi: 10.1109/CVPR.2007.383255. URL <https://doi.org/10.1109/CVPR.2007.383255>. (Cited on page 18.)

Improving Public Access to Government Documents

In this thesis, we address the challenge of the large-scale publication of Dutch FOIA documents in such a way that they adhere to the common FAIR data principles for digital artifacts. Current publication practices mean that much of the FOIA documents published online do not adhere to these principles, limiting the usefulness of these documents to citizens and researchers alike. Since addressing these problems at the source (i.e., before publication) is currently not a viable solution, this thesis focuses on developing and evaluating methods for the post-hoc repair of such documents.

Part I of the thesis is concerned with developing automatic methods for the page stream segmentation of complex dossiers into individual documents, and redacted text detection to benefit downstream page segmentation and OCR output. In Chapter 2 we create a benchmark for the page stream segmentation task, consisting of two datasets of Dutch FOIA documents annotated with document boundaries, and a set of evaluation metrics. We evaluate four different groups of models in both an in-distribution and out-of-distribution setting, and find that the models based on neural networks that use a binary page-classification scheme perform best. In Chapter 3, we investigate the efficacy of neural image segmentation methods for automatically detecting redacted text in FOIA documents. Our results show that these segmentation methods are highly effective in detecting redactions and also have a low false positive rate on pages not containing redactions. Chapter 4 is concerned with creating the Woogle dataset, a large collection of Dutch FOIA documents, collected through web-scraping, and integrated into a search engine. The creation of this dataset shows that using our methods and a strict metadata scheme results in a high-quality dataset that is searchable and more closely adheres to FAIR data guidelines.

Part II of the thesis is concerned with investigating evaluation metrics for the extreme clustering tasks from Part I. In Chapter 5 we revisit the BCubed clustering metric and propose the ELM metric, a slight variation of the original metric that addresses some of the shortcomings of BCubed. Through theoretical proofs and experiments, we show that the ELM metric behaves similarly to BCubed, can achieve a score of zero (unlike the original), and still satisfies formal constraints set out by Amigó et al. [3]. Chapter 6 discusses the Panoptic Quality metric from Computer Vision, and proposes an alteration of this metric that uses a more general matching condition for aligning reference and hypothesized segmentations. Through theoretical proofs and experiments on three image segmentation datasets, we show that the metric behaves similarly to the original, while yielding more true positives. Chapter 7 compares three different metric groups and their suitability for evaluating text segmentation tasks. Through theoretical arguments and experiments on synthetic and real-world datasets, we show that the group of metrics measuring performance on the segment level has the most desirable properties, particularly the overlap-weighted F1 score (the PQ metric), which is well-behaved under different circumstances.

Overall, this thesis has aided in the development of automatic techniques for processing FOIA documents, showing that with some effort, it is possible to develop enabling technology for enhancing the quality of such documents.

Improving Public Access to Government Documents

In dit proefschrift behandelen we de uitdaging van de grootschalige publicatie van Nederlandse Woo documenten op een dussdanige manier dat deze voldoen aan de bekende FAIR data standaarden voor digitale objecten. De huidige publicatiewijzen hebben als gevolg dat veel van de online gepubliceerde Woo documenten niet aan deze standaarden voldoen, met als gevolg een beperkte bruikbaarheid van deze documenten voor zowel burgers als onderzoekers. Omdat het bij de bron aanpakken van deze problemen (i.e. voor publicatie) momenteel geen haalbare oplossing is, richt dit proefschrift zich op het ontwikkelen en evalueren van methodes om zulke documenten achteraf te repareren.

Deel I van het proefschrift is gericht op het ontwikkelen van automatische methodes voor de pagina segmentatie van complexe dossiers in individuele documenten, en het detecteren van geredigeerde tekst, ten behoeve van het verbeteren van latere pagina segmentatie en de output van OCR systemen. In Hoofdstuk 2 creëren we een dataset voor de page stream segmentation taak, bestaande uit twee datasets van Nederlandse Woo documenten geannoteerd met documentgrenzen, en een set van evaluatiemethodes. We evalueren vier verschillende model groepen in zowel de in-distributie en buiten-distributie scenario's, en zien dat neurale modellen die een binaire pagina-classificatie aanpak hanteren de beste resultaten opleveren. In Hoofdstuk 3 onderzoeken we de doeltreffendheid van neurale visuele segmentatiemethodes wanneer deze worden gebruikt voor het automatisch herkennen van geredigeerde tekst in Woo documenten. Onze resultaten laten zien dat dit type segmentatiemodel zeer effectief is in het herkennen van geredigeerde tekst, en dat tegelijkertijd het aantal vals-positieven op pagina's zonder redacties zeer laag is. Hoofdstuk 4 richt zich op de creatie van de Woogles dataset, een grote collectie van Nederlandse Woo documenten, verzameld door web-scraping en geïntegreerd in een zoekmachine. De creatie van deze dataset laat zien dat, door gebruik te maken van de ontwikkelde methodes en een strikt metadata schema, een dataset van hoge kwaliteit kan worden gemaakt, die beter doorzoekbaar is en beter aansluit bij de FAIR data principes.

Deel II van het proefschrift beschrijft het onderzoek naar de geschiktheid van verschillende evaluatiematen voor gebruik met de extreme clustering taken uit Deel I. In Hoofdstuk 5 kijken we opnieuw naar de BCubed evaluatiemaat, en stellen we de ELM maat voor, een kleine aanpassing aan de originele evaluatiemaat die sommige tekortkomingen van de originele maat repareert. Door middel van theoretische bewijzen en experimenten laten we zien dat de ELM maat vergelijkbaar gedrag vertoont aan de BCubed maat, een score van nul kan behalen (in tegenstelling tot de originele maat) en dat het nog steeds voldoet aan de formele eisen opgesteld door Amigó et al. [3]. Hoofdstuk 6 behandelt de Panoptic Quality evaluatiemaat, afkomstig uit het vakgebied van Computer Vision, en stelt een aanpassing aan deze maat voor die gebruikt maakt van een algemenere matching conditie om gold standard en voorspelde clusters aan elkaar te koppelen. Door theoretische bewijzen en experimenten op drie grote segmentatie datasets laten we zien dat de maat vergelijkbaar werkt met de originele maat, terwijl het meer True Positives genereert. Hoofdstuk 7 vergelijkt drie verschillende paradigma's van evaluatiemethodes, en hun toepasbaarheid voor tekst segmentatie taken. Gebruik-

makend van theoretische argumenten en experimenten op synthetische en echte datasets laten we zien dat de groep maten die modelkwaliteit op het segmentniveau meet de meest wenselijke eigenschappen heeft, met name de overlap-gewogen F1 score (de PQ maat), die het best gedrag vertonen onder verschillende omstandigheden.

In conclusie, dit proefschrift draagt bij aan de ontwikkeling van automatische technieken voor het verwerken van Woo documenten, en heeft laten zien dat, met enige inspanning, het mogelijk is om technologieën te ontwikkelen die de kwaliteit van Woo documenten kunnen verbeteren.