

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Information Extraction and Search in Newspapers and Magazines Front Pages

Tiago Silveira Varela



Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Sérgio Sobral Nunes

July 18, 2013

Information Extraction and Search in Newspapers and Magazines Front Pages

Tiago Silveira Varela

Mestrado Integrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

Chair: Professor Jorge Alves da Silva

External Examiner: Professor João Miguel da Costa Magalhães

Supervisor: Professor Sérgio Sobral Nunes

July 18, 2013

Abstract

The research on *content-aware* newspaper archives is still relatively small. This study analyses existing newspaper archives and reviews the state of the art regarding page segmentation methods, text recognition in documents and web search interfaces. This study also details a new keyword extraction method that specifically targets current newspaper front pages. This new method is used to extract information from a collection of newspaper front page images, retrieved from the *Banca de Jornais SAPO* between December 29, 2011 and April 18, 2013. Finally, and using the previously extracted information, we show a prototype for a content-aware newspaper archive.

The analysed newspaper archives usually have simple listing and searching capabilities. These simple features detract users from using them for more than simple visualization of the current day's headlines. Few extract extra information from the newspaper image, i.e. are not content-aware. Those that are content-aware, allow their users to browse and interact with more than just information about the issue date and publication name, e.g. allowing the users to search for keywords or expressions that were actually on the newspaper.

Current page segmentation methods are able to correctly separate the regions of a document, however, they usually target older and stricter document layouts. Modern newspapers have flexible layouts, with information scattered throughout the page. Nowadays, a single issue can have multiple fonts, with varying font-sizes and colours, and their news items are, many times, divided only by coloured regions.

The proposed keyword extraction method splits the newspaper front page image into rectangular regions and applies a background/foreground segmentation method that specifically targets the contained text. Using a test bed designed specifically for this study, this method obtained an 8.62 percentage point increase in accuracy over the default keyword extraction method.

The collection of current newspaper front pages, one of the main motivations for this study, contains 11,728 images, corresponding to 3,842 unique issues from 36 different publications. The information from this collection was retrieved using the developed keyword extraction method and stored in an index, to be accessed through a search interface. This interface, a prototype content-aware newspaper archive, aptly named *Gazeta*, allows the users to search for keywords and to view the front pages containing those particular keywords. It also graphically displays the number occurrences for the specified keywords, over an inserted timespan.

Resumo

Ainda existem relativamente poucos arquivos de jornais que contenham informação extra acerca do conteúdo das imagens que armazenam. Neste estudo é feita uma análise dos arquivos de jornais existentes e é também revisto o estado da arte relacionado com a segmentação de páginas, reconhecimento de texto em documentos e interfaces de pesquisa *web*. Este estudo também detalha um novo método de extração de palavras-chave, destinado especificamente para capas de jornais. Este novo método é também usado para extrair informação de uma coleção de capas de jornais e revistas, recolhida através do serviço *Banca de Jornais SAPO* entre 29 de Dezembro de 2011 e 18 de Abril de 2013. Por último, e usando a informação previamente extraída, demonstra-se um protótipo de um arquivo de jornais que contém informação extra associada a cada capa.

Por regra geral, os arquivos de jornais analisados têm apenas simples capacidades de pesquisa e listagem. São poucos os arquivos que extraem informação extra acerca das capas que colecionam. Os que o fazem permitem que os seus utilizadores pesquisem por palavras-chave ou expressões presentes na capa do jornal.

Os métodos atuais de segmentação de páginas conseguem separar corretamente as regiões de um documento, no entanto estes métodos focam-se principalmente em *layouts* mais antigos e mais rígidos. Os jornais modernos têm *layouts* flexíveis, com a informação distribuída ao longo da página. Hoje em dia, uma única edição pode conter múltiplos tipos de letra, diferentes tamanhos e cores, as notícias estão também muitas das vezes separadas apenas por regiões coloridas.

O método proposto, para extração de palavras-chave, divide a capa em regiões retangulares e aplica-lhes um algoritmo de binarização que se foca especialmente no conteúdo textual das imagens. Usando uma bateria de testes, desenvolvida especialmente para este estudo, podemos comprovar que o novo método obteve um aumento, na precisão da deteção das palavras-chave, de 8.62 pontos percentuais em relação à utilização apenas de um método comum de extração palavras-chave.

A coleção de capas de jornal contém 11.728, correspondendo a 3.842 edições distintas provenientes de 36 diferentes publicações. Usando o método proposto a informação da coleção foi extraída e armazenada num índice de pesquisa. Foi desenvolvido um interface, chamado *Gazeta*, para visualizar e navegar a informação armazenada. Este permite que os utilizadores procurem por palavras-chave e ver as capas que as contêm. Permite também visualizar graficamente o número de ocorrências das palavras-chave ao longo de um período de tempo.

Acknowledgements

Thanks to Professor Sérgio Nunes for all the wise and sensible advice for this study, and constant motivation during the difficult writing part.

To everyone at *Labs SAPO/U.Porto* thanks for providing a friendly and inspiring work environment, and for partially financing this study.

For all the friends, who were there during this long adventure and made it enjoyable and fun, despite all the setbacks and the sleepless nights, my many thanks. Without all of you this would be very different, for the worse.

For my family, supporting me through all of these years, despite not really knowing where I was going, my deepest thanks.

Last, but certainly not least, thanks to Raquel for the much needed motivation and inspiration.

Tiago Varela

Contents

1	Introduction	1
1.1	Problems	1
1.2	Motivation and Objectives	2
1.3	Document Structure	2
2	Newspaper Archives	5
2.1	Global Search	5
2.2	Content Search	6
2.3	Conclusion	7
3	Information Extraction from Front Pages	9
3.1	Image Processing Overview	9
3.2	Page segmentation	10
3.3	Picture Detection	14
3.4	Face Identification	15
3.5	Conclusion	15
4	Search in a Collection of Front Pages	17
4.1	Indexing	18
4.2	Ranking	19
4.3	Interface and Visualization	19
4.4	Visual Cues	22
4.5	Conclusion	22
5	Newspaper collection	23
5.1	Collection Properties	23
5.2	Information Extraction	24
5.3	Conclusion	25
6	newspaper-extractor	31
6.1	Purpose	31
6.2	Algorithm	31
6.3	Annotated Sample	35
6.4	Results	36
7	Gazeta	43
7.1	Purpose	43
7.2	Architecture	44

CONTENTS

8 Conclusion	49
8.1 Future Work	50
References	51
A Sample Results Appendix	55
A.1 Sample Properties	55
A.2 Sample Accuracy Results	56
A.3 Binarization Visual Comparison	60
A.4 Complete annotated sample.	68

List of Figures

1.1	Current front pages from Portuguese daily newspapers.	4
2.1	Newspaper archives — screenshots.	8
3.1	Newspaper from the Google Newspaper Archive.	11
3.2	ICDAR [1] [2] sample images.	12
3.3	BESUS [11] sample and result image.	12
4.1	Faceted metadata with Business Intelligence aggregations [4].	18
4.2	Newsmap website showing a <i>treemap</i> with news.	20
4.3	In The News interface.	21
4.4	Stamen’s In The News providing an overview of the 2004 USA presidential election.	21
5.1	Number of files retrieved per day.	23
5.2	Samples of irregular text quality from the <i>Metro - Grande Lisboa</i> newspaper.	24
5.3	Collection information extraction.	25
5.4	Face detection experiments using the OpenCV library.	27
5.5	Publications issue count along time.	28
6.1	Artefact flow for the newspaper-extractor application.	32
6.2	Front page external contours.	32
6.3	Detecting external rectangular regions from a front page.	33
6.4	Binarization example with FBCI.	35
6.5	Another binarization example with FBCI.	36
6.6	Remaining news items and resulting final output.	38
6.7	Global binarization method.	39
6.8	Comparison between method variants.	40
6.9	Method variants impact on the major publications.	40
7.1	Occurrences for <i>Boston</i> and <i>maratona</i> for April 2–18, 2013.	43
7.2	Keywords with the most number of occurrences for January, 2012.	44
7.3	Request flow for the Gazeta prototype.	45
7.4	Occurrences for <i>saúde</i>	46
7.5	Limiting the date range.	47
7.6	Newspaper issue detail.	47
7.7	Interface with multiple keyword search.	48
7.8	Top keywords for the entire newspaper collection.	48
A.1	<i>Correio da Manhã</i> different binarization methods.	61
A.2	<i>Destak</i> different binarization methods.	62

LIST OF FIGURES

A.3	<i>Diário de Notícias</i> different binarization methods.	63
A.4	<i>Jornal i</i> different binarization methods.	64
A.5	<i>Metro - Grande Lisboa</i> different binarization methods.	65
A.6	<i>Público</i> different binarization methods.	66
A.7	<i>Jornal de Notícias</i> different binarization methods.	67

List of Tables

2.1	Newspaper archives — capability grid.	7
5.1	Collection and selected publications width properties, in pixels.	24
5.2	Collection and selected publications height properties.	25
5.3	Collection and selected publications size properties, in kilobytes.	26
5.4	Publication issues count.	27
5.5	Collection and selected publications rectangular regions count.	29
6.1	Accuracy properties per method variant.	41
6.2	Number of incorrect words per method variant.	41
A.1	Sample publications width properties, in pixels.	55
A.2	Sample publications height properties, in pixels.	55
A.3	Total number of words detected, for the issues, per method variant.	57
A.4	Number of correct words, for each issue, per method variant.	58
A.5	Number of incorrect words, for each issue, per method variant.	59

LIST OF TABLES

Abbreviations

CSV	comma-separated values
DPI	dots per inch
FBCI	Font and Background Color Independent
HCI	Human-computer Interaction
ICDAR	International Conference on Document Analysis and Recognition
NCUT	Normalized Cuts
OCR	Optical Character Recognition
XML	Extensible Markup Language

Chapter 1

Introduction

Newspaper readership has declined in the past few years. In 2012, on the period of January to April, daily Portuguese newspapers sales dropped 11.5%, this equals to 29 thousand less newspapers sold during the same period in 2011. Despite this, daily newspapers and magazines circulation values are still high with the market leader, *Correio da Manhã*, selling on average 115.821 daily copies between the same period [35]. Also, free distribution newspapers like *Metro* and *Destak* have high circulation, and sometimes even higher readership values on the metropolitan areas. This may be because people usually prefer to read using a paper format, despite the advantages of the digital format [23].

Nevertheless, the future is moving towards the increase of digital versions of newspapers and magazines. However front pages, reporting the most important news headlines and some content, are still used on digital publications. There are free newspaper archives, like *Newseum's Today Front Pages* [31] or *Kiosko.net* [25], that gather daily newspaper front pages but the search options are very limited, allowing only to view the latest editions and to search only by date or country.

The purpose of this study is to improve information extraction methods by targeting specifically newspaper front pages, and to create a *content-aware* newspaper and magazine archive containing the extracted information—a web application that is able to not only display newspapers and magazines front pages but also allow users to search for keywords or expressions that were present in those front pages.

1.1 Problems

The most challenging part of this study is the correct extraction of information from the newspapers and magazines front pages. This is due to the fact that these front pages do not have a strict layout and the information is scattered all over the page, e.g. with different fonts and font sizes, text directly over images without a background framing it, text lines wrapping contoured images. This poses serious challenges when trying to properly identify and extract the information. Figure 1.1 displays 4 newspaper issues, from daily Portuguese publications, and the diversity of their

layout arrangements is clear. Also, different news titles and body text don't keep the same size, font or colour for the same issue.

A different group of challenges emerges when trying to show the user all of this information. Search user interfaces often have cluttered interfaces that detract from a good user experience with the system. Trying to pleasantly represent all the extracted information and provide a comfortable and easy way to use the interface is in itself a large challenge and an important part of this work.

1.2 Motivation and Objectives

This work is being developed in collaboration with *SAPO* [38], which has a large collection of newspapers and magazines front pages from their service *Banca de jornais SAPO* [37].

After searching for existing newspaper front page archives it was evident that most of these had only simple search and listing capabilities. Also, the interfaces of these systems were not good and that decreased the user interest in actually spending time browsing or searching for information.

It would then be interesting to create a system, that could automatically extract the information from the newspapers front page images and, not only provide rich information about the content of these front pages, but also display that information in such a way that inspired the user to look for it or to explore related information.

The objectives of this study are the analysis of information extraction methods from image documents and the implementation of an information extraction method, that targets current newspaper front page layouts, using parts from existing methods. Adjusting the parameters of that method to better extract information from the specific collection of newspapers and magazines front pages is also a component of this study. Finally, the last objective is to study the extracted information, and index it, in order to create a search and browsing interface that is easy to use and that actually encourages users to look for content.

1.3 Document Structure

This document is divided into eight Chapters, where:

Chapter 2 — Newspaper Archives

gives an overall review of the existing newspaper archives, comparing their searching capabilities and usability.

Chapter 3 — Information Extraction from Front Pages describes the first problem when building a content-aware newspaper archive—keyword extraction. This Chapter gives a quick overview of some image processing operations and more specific methods for document segmentation.

Chapter 4 — Search in a Collection of Front Pages

describes the problems of indexing and visualizing the extracted information. This gives

Introduction

an overview of how to store the extracted information in an index, what methods are available to quickly find related information and some existing interesting keyword and news visualization methods.

Chapter 5 — Newspaper collection

describes the newspaper and magazine collection that was available for this study and analyses some of its properties, including examining the script that was used to extract information from the collection files.

Chapter 6 — newspaper-extractor

details the *newspaper-extractor*, the keyword extraction application developed during this study, as well as describing the methodology used to evaluate the accuracy of the method.

Chapter 7 — Gazeta

describes the *Gazeta* prototype web application, a content-aware newspaper archive, created to demonstrate some the uses for the extracted information.

Chapter 8 — Conclusion

wraps up this document, detailing the conclusions and future work for study.

Introduction

Tinham seis escravos e recebiam Rendimento Mínimo
Valongo A sogra Carminda, o genro Málio Fernando e o capitaz Ludovino escravizaram ucraniano, deficiente, sem dedos, epiléptico, sem-abrigo, demente e inválido, andavam de Mercedes e tinham contas bancárias em Espanha. No



(a) *Jornal de Notícias.*



(b) *Público.*



(c) *Correio da Manhã.*



(d) *Jornal i.*

Figure 1.1: Current front pages from Portuguese daily newspapers.

Chapter 2

Newspaper Archives

There are many websites that store images of newspapers and magazines. Some only keep front pages while others keep the complete issue. Most provide only simple search capabilities, e.g. to find newspapers for a given date or to find newspapers or magazines from a specific country or state. Few provide more advanced search capabilities like searching for keywords inside a newspaper issue.

2.1 Global Search

This Section describes archives that provide only simple listing capabilities or global search, i.e. the ability to search only attributes of the entire collection—such as the newspaper title or theme—instead of being able to search the contents of each issue or page. These archives have limited value when trying to search for particular news items, events or personalities.

The *Today's Front Pages* section of the *Newseum* website [31] provides each day over 800 newspaper front pages, 7 of them Portuguese newspapers. However, due to copyright issues, they only store older front pages of events with historical significance. There are no search capabilities, and the user can only: list the newspapers by region, view a gallery with small images of the front pages and the name of the corresponding newspaper, or view the newspaper distribution using a map. Figure 2.1a displays a screenshot with this archive's index page.

Kiosko.net [25] provides many daily newspaper covers from various countries, including 5 Portuguese daily newspapers and 2 Portuguese sports newspapers. Figure 2.1b shows a screenshot from this archive displaying the European front pages. However, it only keeps in archive front covers from the latest 7 days. Their intention is to give a “quick overview of current events” and they do it by providing a gallery that displays thumbnails of the newspapers from different countries. It does not provide searching capabilities beyond filtering by region or country.

Banca de jornais SAPO [37] is a Portuguese newspaper archive which allows access to the front pages of national and international newspapers and magazines with broad themes such as sport, financial, technology, and daily newspapers. It only allows access to the current day's front

pages. It also only allows the user to search by newspaper or magazine names, or to search by theme. Figure 2.1c show a screenshot of the archive displaying an enlarged front page.

Mário Soares Foundation provides access to *Diário de Lisboa Online* [3], an archive that stores all editions with full content of the *Diário de Lisboa* newspaper over its circulation span, from 1921 to 1990. The scans are treated to provide a clear segmentation of the background from the foreground, i.e. black text over clean white background. However it only allows the user to navigate from issue to issue and inside each issue from page to page—search functionalities do not exist. Figure 2.1d is a screenshot of this archive, displaying the issue information on the left, the scanned image on the centre, and on the right side has buttons for issue navigation and zooming.

2.2 Content Search

The archives in this Section treated their newspapers in such a way that they were able to extract some of the information from them. In two of the described cases the methods used to extract the information were not publicly available. This treatment allows the user to search for content that is inside the newspaper image, e.g. the title or the body text of a news item.

Google News website has a section about old newspapers [19] which provides listing and navigation through the full newspaper content. It has news segmentation, allowing the user to link directly to a particular news post from a newspaper. However it does not allow the user to search the contents of a newspaper and the global search includes results that are not exclusive to the newspaper archive. On May 2011, this project was discontinued, meaning that no new content would be scanned [20]. Figure 2.1e displays a screenshot of the archive, with a detected news title highlighted. The method used to extract the information for this project is described in Section 3.2.

NewspaperArchive is a paid subscription website [32] that keeps mostly newspapers from the USA, Canada and United Kingdom. It allows the user to browse newspaper pages, to easily, yet inaccurately zoom in and out onto a page and, the most interesting feature, to search a newspaper using keywords, enabling the user to find content that is present on the newspaper's news. After a user searches for a particular keyword the system highlights it on the displayed page. Figure 2.1f is a screenshot of the application, viewing a single newspaper, and shows the zooming and keyword highlighting capabilities.

The city council of Arganil provides an archive of an old weekly newspaper — *A Comarca de Arganil*, containing the first series from 1901 to 2009 [13]. The newspapers are scanned and treated to detect words. They show the original newspaper image with the text colour treated to provide better contrast. There is an advanced search feature that allows the user to search by keywords, date, edition, or to provide a issue number range and view the issues available. This search by keyword allows the users to search for news content inside any of the newspapers. Despite these good features it has some bad user interface practices: not allowing the user to browse the issues easily, only if a user searches by date or using an interval of dates; it allows the user to zoom the pages when browsing an issue using double click, however it forwards the user to a new page

view with only zooming capabilities, i.e. losing all issue browsing and content search capabilities. Figure 2.1g shows a screenshot of this newspaper archive displaying a front page and the advanced search form.

2.3 Conclusion

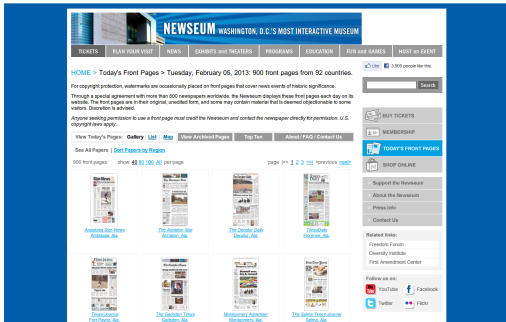
As described above, many of these newspaper archives only provide simple search functionalities, e.g. searching by newspapers from a specific source or from a specific country. More advanced search functionalities, e.g. searching inside a specific newspaper, are provided only by a few of the archives.

Those, who have content-aware search, do it in unusual ways, e.g. the *Google News* newspaper archive, which allows an user to search for keywords that were detected while processing the newspaper, but does not provide the user with a search bar that only finds keywords in the page, or issue, that the user is currently viewing. Table 2.1 provides an overview of the capabilities and some details of each archive. Figure 2.1 displays screenshots from all of these newspaper archives—to provide an overall feel of their user interface and functionalities.

Table 2.1: Newspaper archives — capability grid.

	content-aware	single issue search	gallery view	page pre-processing	zooming capabilities	complete issues	paid service
Newseum	—	—	✓	—	—	—	—
Kiosko	—	—	✓	—	—	—	—
Banca de Jornais SAPO	—	—	✓	—	✓	—	—
Diário de Lisboa Online	—	—	✓	✓	✓	✓	—
Google newspaper	✓	—	✓	✓	✓	✓	—
Newspaper Archive	✓	✓	?	✓	✓	✓	✓
A Comarca de Arganil	✓	✓	—	✓	✓	✓	—

Newspaper Archives



(a) Newsseum Today's Front Pages.



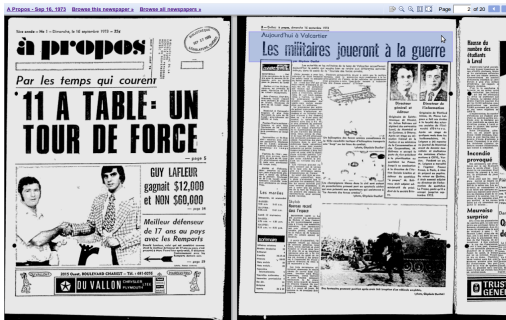
(b) Kiosko.



(c) Banca de Jornais SAPO.



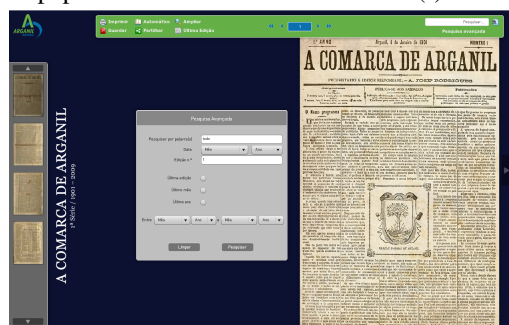
(d) Diário de Lisboa Online.



(e) Google News newspaper archive.



(f) NewspaperArchive.



(g) A Comarca de Arganil.

Figure 2.1: Newspaper archives — screenshots.

Chapter 3

Information Extraction from Front Pages

Correctly extracting information from the newspapers and magazines front pages is a crucial part of the content-aware archive this study aims to build. One of the goals of this study is to develop a method that specifically targets current newspaper front page layouts and that is able to extract information from those images. To accomplish that, the front pages should first be treated and then analysed properly. This Chapter provides a quick overview of some image processing methods and detailed reviews of page segmentation, face detection and picture extraction methods.

3.1 Image Processing Overview

There are several ways to process image files in order to extract information from them. This section provides a quick overview about *binarization* methods and *morphological transformations*.

Binarization, also known as thresholding, is an image processing method that turns a colour or, more commonly, a greyscale image into a binary image, i.e. an image with only two colours, usually pure black and white. This works by comparing each of pixels in an image with a given value, i.e the threshold, and replacing it with white colour if it exceeds the threshold or black if it does not [6, Chapter 5].

Morphological transformations are operations on an image with the purpose of reducing noise, thickening or thinning lines or joining disconnected elements. The most basic morphological transformations are *dilation* and *erosion*. The dilation operation can be thought as a local maximum of each pixel in a image with the immediate neighbourhood, i.e. replacing each pixel value with the maximum value between itself and its neighbourhood. This allows the growth of the brighter regions of an image, overlapping their darker neighbour pixels. The erosion operation is the inverse of dilation, it corresponds to a growth of the darker regions in an image [6, Chapter 5].

It's possible to create compound operations by applying these basic operations in sequence. It's called an *opening* operation when we erode first and then dilate. A *closing* operation is when we dilate and then erode [6, Chapter 5].

Using *edge detection* is also a powerful way to process images. These operations simplify the content of an image, reducing the quantity of data to be processed, all while keeping structural information about the content's boundaries [8]. The method developed by Canny, in 1986 and now aptly named the *Canny edge detector*, is still used today to accurately detect edges in images. This method however needs 2 thresholds to be defined by the user of the algorithm, these are called the *hysteresis thresholds*. Canny recommended that the ratio between, the *high* and the *low* thresholds, should be between 2:1 and 3:1 [6, Chapter 6]. Wong suggests that, as a *rule of thumb*, thresholds of 0.66 and 1.33 times the image's mean or median value quickly return an acceptable result [50].

3.2 Page segmentation

There are several methods to segment pages and most of them use background detection followed by Optical Character Recognition (OCR) and block segmentation using gutters and line delimiters.

Google had a *Newspaper Digitization, Index and Search* project where they scanned microfilm pictures of old newspapers, and treated that information in order to index it [9]. Here are described some of the most interesting steps of this process since it had some extra steps that are not essential for this study, e.g. page flip correction or newspaper binding. The first important step is page cleaning where the image gets *binarized*, here meaning that the background is painted white and the foreground value is mapped to one of four values at the dark side of the spectrum. This gradient is done to avoid aliasing artefacts. The following step is to use OCR. They use a third party method that despite being "one of the leading OCR engines in the world" still makes many mistakes. These are probably due to the high level noise and multiple font variations in a single page. One mistake that they mention is that sometimes large headlines are recognized as pictures, their turn around was to remove the OCR detected text from the page, scale down the page and OCR again.

Next is block segmentation, where text blocks are identified using gutters and lines on the page. Not much information is provided on how this was accomplished, only that special filters were developed to detect gutters. These filters compute the background pixels neighbouring a specific pixel to determine whether it belongs to a gutter or not. Following this, headline detection is performed where information from the OCR font-size is used to classify the blocks into headlines or body text. The area-perimeter ratio of the connected components is also used as a cue.

The final step is the article segmentation where they use a rule based binary classifier to distinguish if text blocks belong to the same article or are from different articles. There are two dominant rules: Common Headline Rule where text blocks below the same headline belong to the same article and Orphan Block Rule for text blocks with no headline above them, e.g. when a news article is extended into a new column. The segmentation accuracy of this method was of approximately 90% and the OCR accuracy of around 80%. The segmentation accuracy was measured against manually annotated ground truth and the OCR accuracy by the ratio of dictionary words correctly recognized.

Despite the good results described, this project worked only with older newspapers which had a simple layout with text blocks properly organized and without many pictures between them. Because of this the OCR component of the process could be performed in a earlier stage. With current newspaper front page layouts the existing OCR methods have trouble recognizing words due to the disorganized news layout.



Figure 3.1: Newspaper from the Google Newspaper Archive.

The International Conference on Document Analysis and Recognition (ICDAR) had a page segmentation competition with four editions (2003, 2005, 2007 and 2009), organized by the *PRIMA* [34] research lab. The goal of this competition was to evaluate methods of document layout analysis (page segmentation and region classification) [1]. Following is the description of two of the best methods developed for the 2007 and the 2009 competitions.

First, the *BESUS* method, which stands for Bengal Engineering and Science University, Shibpur (India) and was developed by S.P. Chowdhury, S. Mandal and A.K. Das (of that university) in association with B. Chanda of the Indian Statistical Institute (ISI) in Calcutta [1, 11]. Apart from being in the competition in 2007 this method was also submitted for the 2003 and 2005 edition. It was the best overall method of the 2007 competition [1]. This method is divided in four steps, but the first one is not important to this study. This first step is skew correction, i.e. the correct alignment of the digitally scanned page, and the delimitation of the page area since the scan introduces unwanted artifacts to the border of the image. The images we're going to treat have no skew and the full image is the newspaper or magazine area intended to segment.

The second step is graphics segmentation where a greyscale image is created. After this, images and graphics are removed from the page. This is done by using morphological open and

Information Extraction from Front Pages

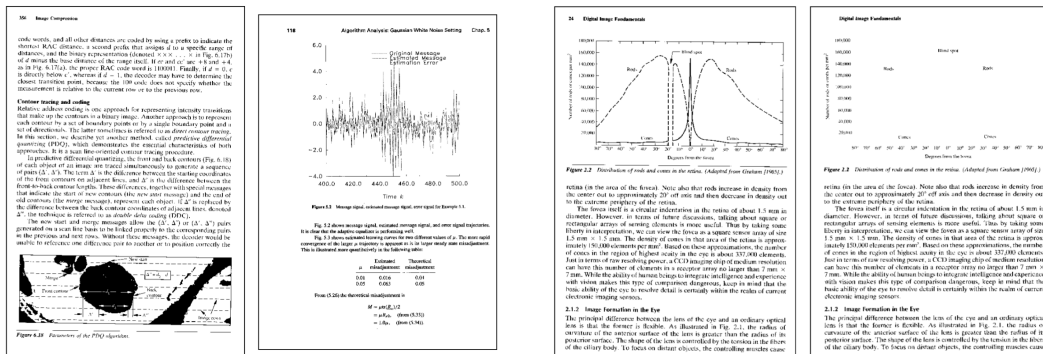


(a) From the 2007 edition.

(b) And the 2009 edition.

Figure 3.2: ICDAR [1] [2] sample images.

close operations which create features that can be analysed concluding if they are images or not. The following step is to segment line art. Regions are computed on all components and groups are formed. After the similarity of the components is examined the line art ones are removed since they exhibit different characteristics from the text lines. After this step what remains on the page is mostly text, however no document structure is obtained, i.e. grouping headlines and body text.



(a) Sample image.

(b) Text detection.

Figure 3.3: BESUS [11] sample image and result image.

The *Fraunhofer Newspaper Segmenter* was presented by Iuliu Konya, Stefan Eickeler and Christoph Seibert from the *Fraunhofer Institute for Intelligent Analysis and Information Systems* at Sankt Augustin, Germany for the ICDAR 2009 Page Segmentation competition. Their method was the best overall from competition and deserves a more in-depth description, although only limited information is available [2]. This method starts with a pre-processing part where a global optimal binarization algorithm is applied to the input image, which is in greyscale format.

The next step is to detect the black separators, i.e. the lines between news or the ones that

separate the pictures from the news title or body. They start by improving the quality of the separators using the method developed by Gatos et al. [17] where line breaks or dotted lines are connected using a set of morphological operations on the binary image. After this, the separators are extracted and a triage is made using information about the character size that is mostly used on the page. The third step is to detect white separators, i.e. the empty space between news or pictures. This step uses the algorithm described by Breuel [7] to detect maximally empty rectangles. These detected rectangles have to pass through an approval stage where they must meet certain conditions like having a large enough height in relation to the dominant character size.

The fourth step is to segment the page. Here a hybrid approach is used, based on Jain and Yu's method [23], consisting of a bottom-up process where pixels are grouped into nodes, nodes into connected components and horizontal or vertical lines, connected components into text lines and text lines into region blocks. This bottom-up process is guided by top-down information given by the column layout of the page, obtained through the list of separators [2]. Also in this step text regions are detached from non-text regions using statistical characteristics of text, e.g. characters aligned on lines. The fifth and final step is where exact text lines are detected again from the text regions obtained after step four. This uses a method similar to Jain and Yu's [23]. Here the characteristics of the font, e.g. stroke width, x-height¹, italics, are used to obtain text regions with similar properties [2].

The paper by Grover et al. [21] describes a technique to extract text from a document image based on the sharp edges that text characters have. Their method starts by converting the document image to greyscale and then performing edge detection using a *Sobel* operator, keeping only hard edges and eliminating the weak ones using a thresholding operation. Next, they divide the image into a grid of small non-overlapping blocks, whose edge size is related to the image resolution. This global edge size can oscillate between $0.05 \times R$ and $0.3 \times R$ with R being the image resolution in dots per inch (DPI)—their experiments showed that an edge value between $0.1R$ and $0.15R$ was better suited for smaller text and between $0.2R$ and $0.25R$ for larger text.

The following step is to classify the blocks into text or non-text blocks. They calculate the average gradient magnitude per edge and per pixel of each block and given a certain threshold it divides the blocks into two groups. This classification is based on the fact that “the number of edge pixels and their gradient magnitudes are usually higher for text than for non-text blocks” [21].

This method was tested on images of magazines and newspapers, with different background colours, and the results show a sensitivity of around 99% and a false alarm rate of around 5%. However, the method is not totally automatic, requiring manual adjustment of the block edge size, the threshold to remove weak edges and the threshold to classify the blocks into text.

¹The height of the lowercase letter x in the particular font.

3.3 Picture Detection

Picture detection in documents is a difficult problem, particularly if the documents are magazines or newspapers with complicated and overlapping graphic designs [10]. While this study will not focus in this field, it could be useful to optimize the face detection part of the work—reducing the search area.

Chiu et al. [10] present a method to detect pictures in technical papers and reports, which uses OCR to separate the text from the rest of the elements and applies an algorithm to cluster the remaining components of the document into possible image regions, i.e. photos, block diagrams, graphs and screenshots.

Their method starts by using *Microsoft Office Document Imaging OCR* to extract words and their corresponding bounding box. Once this is done they fill the bounding box with the document background colour effectively removing the detected text from the document and returning a document with only image regions and some artifacts, e.g. horizontal or vertical lines. Since some of the documents have high resolution, and that is not necessary for the following steps, they downsample the resulting image, taking care to *smooth* the image before downsampling to avoid aliasing artifacts, i.e. jagged lines. Once this is done they convert the downsampled image to a binary image using a low threshold, effectively colouring everything that isn't background with the same colour. The following step here is to keep only the edges of the image blocks, for improved efficiency, using an edge detector algorithm. After this, they remove image blocks whose width and height are below a certain value in order to clean the image to remove small specks and noise. Also, components with a high aspect ratio are removed, thus deleting from the document the separator lines. Next, they use the Normalized Cuts (NCUT) algorithm [30] to cluster the possible image regions together. According to the article, NCUT could be used directly on the document image, with good results, for photograph segmentation, however it would fail to recognize image blocks that were not as dense in terms of colour or intensity, e.g. line graphs or block diagrams.

The final step of the article's method is meant to improve the cluster classification and consisted on using the OCR detected captions and counting the "Figure" keywords present there, to have a better understanding of how many images should the clustering algorithm detect. This final step is not particularly interesting for this study since newspaper and magazine images are generally not labelled.

To assess the quality of the method, the pictures of 2 sets of random papers were manually annotated. The first set contained 10 technical papers from 1988–1997 and the second set contained 20 papers from 2009. In the first set the method correctly detected 80% of the pictures with a false alarm rate of around 5%. For the second set, the accuracy was around 87% and the false alarm rate was about 10%.

3.4 Face Identification

An interesting way to extract more information from front pages would be to detect and recognize the faces of people that were present on the news pictures. Using *OpenCV* [44], a computer vision library, face detection is a relatively quick and simple operation [42, 43]. According to its documentation, it uses features detection to find objects in images. These features work by detecting “oriented contrasts between regions in [an] image” and, using a previously trained set of features, can be used to detect “contrasts exhibited by a human face and their spacial relationships”, allowing then to detect traits of human faces in images [43].

Face recognition is a more expensive task, requiring training of each individual face using several images, more than 6, to accomplish accurate results, upwards of 90% [41]. However, this study will not focus on face recognition.

3.5 Conclusion

After reviewing the page segmentation methods it is difficult to assess which one is better suited to break current newspaper front pages into manageable components.

This is due to the fact that most of the methods described treated only older newspapers with stricter layouts or documents with different nature, with specific layouts, such as technical papers. It’s also not completely clear if a single method is the better approach here or if a combination of different parts from each method would yield better results.

Chapter 4

Search in a Collection of Front Pages

Displaying information in an appealing way is an important part of the prototype, content-aware, newspaper archive that this study aims to create. Fry [16], in his thesis entitled “Computational Information Design”, describes the steps required to understand and display data, transcribed here:

1. **acquire:** the matter of obtaining the data, whether from a file on a disk or from a source over a network.
2. **parse:** providing some structure around what the data means, ordering it into categories.
3. **filter:** removing all but the data of interest.
4. **mine:** the application of methods from statistics or data mining, as a way to discern patterns or place the data in mathematical context.
5. **represent:** determination of a simple representation, whether the data takes one of many shapes such as a bar graph, list, or tree.
6. **refine:** improvements to the basic representation to make it clearer and more visually engaging.
7. **interact:** the addition of methods for manipulating the data or controlling what features are visible.

The described steps can be translated to this study. Once enough information is extracted from the front pages—step 1—it needs to be stored and categorized properly to allow easy access to it and also to its related information. This involves the development of a search engine with an index to properly organize and store all the information—steps 2–4—and the creation of a usable interface for the user to be able to interact with the system—steps 5–7. This Chapter describes interesting concepts to keep in mind while developing a web search interface and displays some existing news visualization applications.

4.1 Indexing

To index all the necessary information we will use Apache Solr [15], an open source search platform, which allows us to index the extracted information and to create appropriate categories for it. This platform provides many facilities including full-text search, hit highlighting and category navigation, i.e. faceted search which is also known as guided navigation.

The use of faceted search “provide[s] an attractive interface for exploring multidimensional data” [4]. To achieve this guided navigation, facets relevant to the collection are created, e.g. area of interest or country region. These facets are usually independent of each other but can also be hierarchically linked [51]. After the creation of these, each document in the collection can be assigned any number of these labels [22, Chapter 8]. This allows the user to access the same information through completely unrelated category paths.

Depending on the information available after the page segmentation process, categories can be created using sections of the newspaper, e.g. if a daily newspaper has a sports section on the front page the corresponding news could be labelled as sports news, allowing a user to further drill-down on the information of a newspaper, i.e. move from general content to more specific one [4].

A paper by Ben-Yitzhak et al. [4] shows useful improvements to simple faceted search interfaces. One of those improvements focuses on the fact that simply counting the number of documents in a facet is not helpful in providing valuable information about the facet. They propose an approach closely related to business intelligence where the analysis of the data provides better hints when refining a search query.

Using the examples described in the paper, when browsing for books and trying to refine by author it would be more helpful to list the author by the number of best-sellers books they’ve written or by the number of book copies sold than to simply list by the number of books an author has written; this example is represented in Figure 4.1. Another example used in the paper was that when listing country strength by viewing the company contracts across various countries it would be more helpful to list the countries by average contract values than simply by the number of contracts, because countries with many contracts but of lower values would be given an inaccurate listing advantage when simply listing by the number of contracts.

Narrow search by:						
Author			Rating(worst 1-10 best)		Binding	
	<i>Best sellers</i>	<i>Average Price</i>		<i>Max relevance</i>		<i>Grams per page</i>
James Stewart (5)	3	138.95	10 (557)	0.97	Paperback (2086)	1.72
Chad Fowler (2)	2	38.95	8 (584)	0.84	Hardcover (550)	2.35
James Surowiecki (3)	2	22.15	6 (174)	0.78	Mass Market	0.48
Gary B. Shelly (13)	2	66.33	9 (649)	0.78	Paperback (33)	
Other (4193)	81	40.92	Other (758)	1.6	Spiral-bound (14)	2.6
					Other (39)	2.28

More Facets
[Publication Date](#)
[Category](#)
[Subject](#)
[Publisher](#)
[Salesrank](#)
[Price](#)
[Weight \(grams\)](#)

Figure 4.1: Faceted metadata with Business Intelligence aggregations [4].

The indexing process done by Solr follows a typical *inverted index* where the indexed keywords are associated to the document in question. This means that for each keyword in the index, Solr keeps a *postings list*, which is a list of documents where that keyword appears. Each element of this postings list can also keep the offsets within the document for where that particular keyword appears—these offsets are used for phrase matching and are also a way to determine the relevance of the keyword in a document [4].

4.2 Ranking

The order of the results for a query should follow a ranking process to provide more accurate results. In the traditional sense if a keyword appears in the top of a document or on its title then it should be an important keyword. This is a typical approach in relevance ranking [26]. This analogy can also be made for the particular documents in this study, i.e. the news of a newspaper or magazine, and additional characteristics of the extracted information, e.g. the font size or colour, could be used to possibly increase the ranking accuracy.

Using different weights for distinct parts of the structure of the document would also allow for fine tuning of the query results in the stage of user testing. This means that when running query tests with real users they could also rate the relevance of the results depending on the importance, i.e. weight, given to the title or the body of the news.

4.3 Interface and Visualization

The user interface of a search system, or in fact any system, is a key part for the success of its usage. Since the main purpose of these systems is to provide search and browsing abilities to the user their design should be focused on the human interaction [27].

An important concept to keep in mind, when designing an interface that focuses on human interaction, is *usability*, which qualifies how easy an interface is to use. Nielsen [33] defined five components of usability, transcribed here:

Learnability: How easy is it for users to accomplish basic tasks the first time they encounter the design?

Efficiency: Once users have learned the design, how quickly can they perform tasks?

Memorability: When users return to the design after a period of not using it, how easily can they reestablish proficiency?

Errors: How many errors do users make, how severe are these errors, and how easily can they recover from the errors?

Satisfaction: How pleasant is it to use the design?

However, despite this clear component definition, it is still very difficult to create highly usable interfaces [22].

In his book entitled “The Laws of Simplicity”, Maeda [29] defines ten *laws* and three *keys* to achieve simplicity in the domain of technology. Some of the most interesting laws are to *Reduce*, which is to think of *how simply a feature could be made* and *how complex would it have to be*, and trying to find the correct balance between these two concepts [29, Law 1]; to *Organize*, with the focus here being on the fact that a correctly organized system may appear smaller and simpler to use than a system which labels the same information using a large number of tags [29, Law 2]; the *Time* law states that “savings in time felt like simplicity” and an interesting example is given when displaying a progress bar that increments periodically over time and one that only increments after completion—the bar that increments periodically is *perceived* as faster than the other, and as the author points out to showing “people how much time they have left to wait is a humane practice” [29, Law 3].

Also related to simplicity, but here in the form of visual simplicity is the concept of *non-data ink ratio*, coined by Tufte [47]. He points out that reducing non-data ink, i.e. lines that are not related to data itself, is of great importance when trying to represent information in graphical form. This concept is also reiterated by Fry [16].

While discussing information visualization it’s interesting to review some previous works, about news visualizations, that could be in some way applied to this study.

In 2004, Weskamp [48] created *Newsmap*¹, a website that displays an updated *Google News* feed using a *treemap* representation—the system is displayed in Figure 4.2. This interesting way of

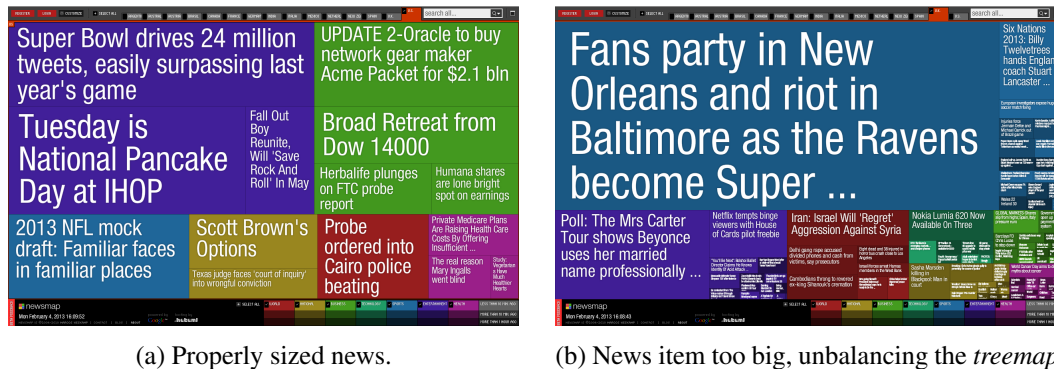


Figure 4.2: *Newsmap* website showing a *treemap* with news.

representing news items uses colours to represent categories and the size of each block is given by the number of related articles, effectively giving more emphasis to stories that had more coverage [48]. It’s a very effective way to have an overall view of the most important news at that moment, however sometimes the generated tree blocks are too small to contain information—this defect is visible in Figure 4.2b.

¹<http://newsmap.jp>

Also in 2004, *Stamen*² created *In The News* [40], an interface that provided visual comparison of specific topics based on occurrences over time. This visualization displayed the news as coloured boxes across a vertical weekday axis. Each colour represented a news topic and the box size showed the topic's relevance for a particular day. Colour hues were used to denote if a particular topic had lost or gained importance from the previous day [49]. Figure 4.3 provides a complete overview of the interface.

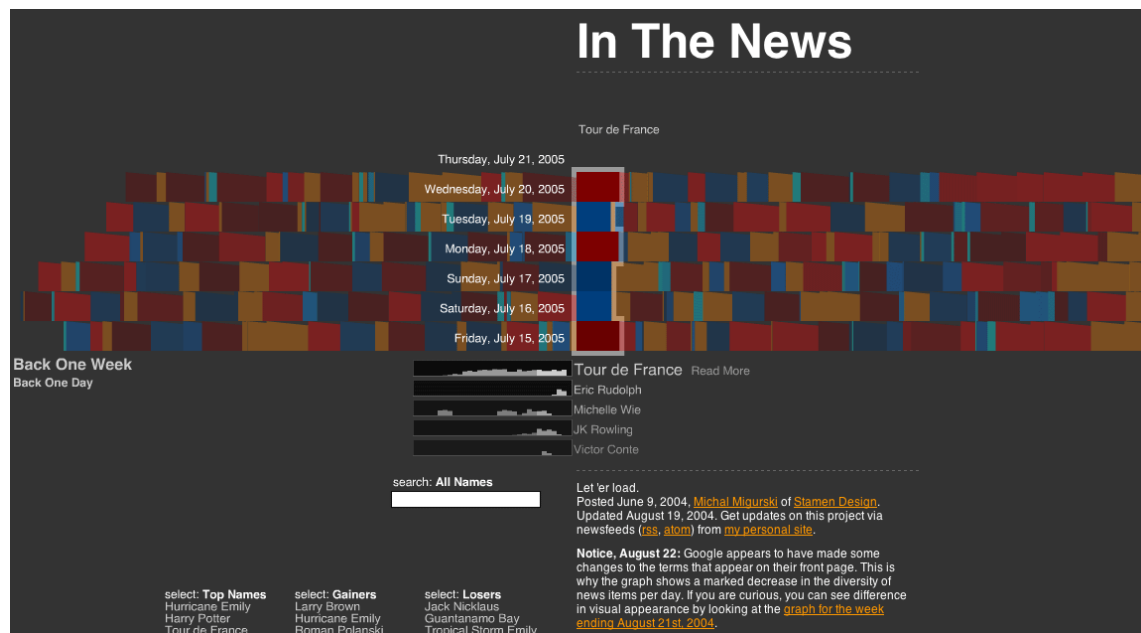


Figure 4.3: *In The News* interface.

It also displayed *sparkline-style* graphs that provided a quick overview on each of the specified topics [40]. The information was initially retrieved from *Google News* and, later on the life of the project, from *Delicious*' most popular items. Figure 4.4 shows a sample of the visualisation interface for the 2004 USA presidential election.

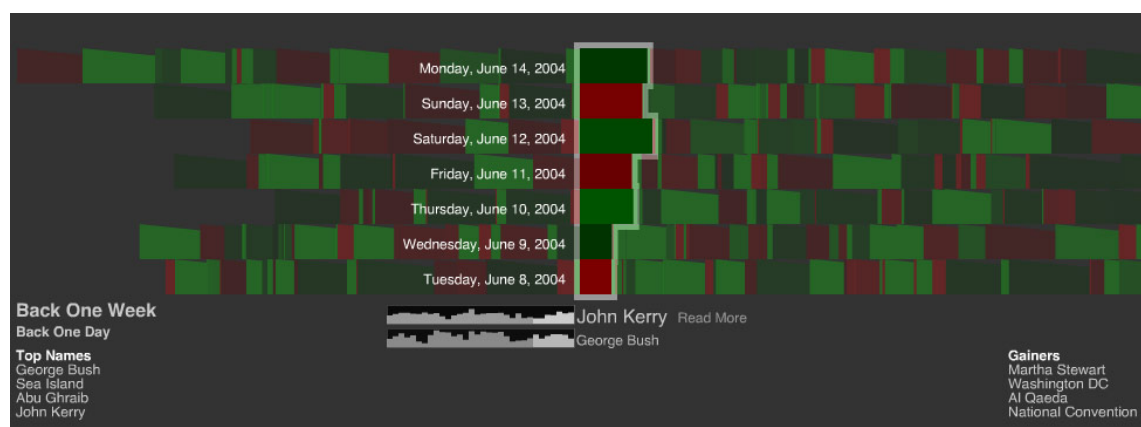


Figure 4.4: *Stamen's In The News* providing an overview of the 2004 USA presidential election.

²<http://stamen.com/>

4.4 Visual Cues

The visualization part of the search engine should keep existing concepts of search user interfaces, e.g. the snippets results providing information relative to a search query on web search engine have been almost unaltered since their inception.

Hearst [22] mentions that this unaltered look is most likely due to the fact that the “[users] do not want their flow of thought interrupted by an intrusive interface” and that “the fewer distractions while reading, the more usable the interface [is]”; she also points out that almost everyone uses search interfaces when browsing the Internet and because of that their design must be easily understandable and able to be used by users with a wide variety of ages, cultures and backgrounds [22, Chapter 1]. It is then useful to keep these visual cues in order to help the user feel comfortable but also introduce some other functionalities like faceted search which are appropriate to allow the user to navigate in large image collections.

In the paper by Yee et al. [51] a usability study was made where 32 art history students were asked to explore a collection of 35,000 fine arts images using a faceted search and a traditional image search interface; 90% of the students preferred the faceted search interface and 97% reported that the interface helped them to learn more about the collection.

Another important cue that helps the user to get some feedback on what he is doing is to show results immediately after the initial query [22, Chapter 1]. This can be achieved through *live search* where the system performs the user query as soon as he has stopped typing and displays the results on the page [39, Chapter 13].

Also mentioned by Scott and Neil [39, Chapter 13] the usage of *live suggestion* where—as the user is typing the query—the system provides keywords that either complete what the user was writing or are related to the query being written.

4.5 Conclusion

This Chapter provides good user interaction principles that one should be aware when developing a search user interface. Even though some concepts appear somewhat abstract, e.g. the *data-ink* ratio and “The Laws of Simplicity”, they are important design pieces that we should be aware in order to provide a simple and clean interface that users would be glad to use.

Although this Chapter mentions that faceted search is very useful when developing a search interface, it would be very difficult to implement it during this study. This is due to the fact that we only possess very little information about the content of the collection. Only the date and the name of the publication for each image. Once we extract the text content from the front page images we could classify the content, but that would involve further processing and work that is not intended for this study.

Chapter 5

Newspaper collection

One of the main motivations for this work was the availability of a large collection of recent newspaper and magazine front pages. These were retrieved, daily, from December 29, 2011 to April 18, 2013, from the *Banca de Jornais SAPO* [37] newspaper archive’s *Nacional* section, i.e. the main section of the archive.

5.1 Collection Properties

The collection contains 11,728 images, distributed over 426 folders. Each folder corresponds to a day’s worth of retrieved newspapers. From this large collection of files, only 4 images were corrupted and unusable for further analysis. Figure 5.1 shows the number of images collected per day. This reveals some gaps where no images were retrieved, noticeably from October 9 to November 4, 2012, where the retrieval system was down due to a regressive update.

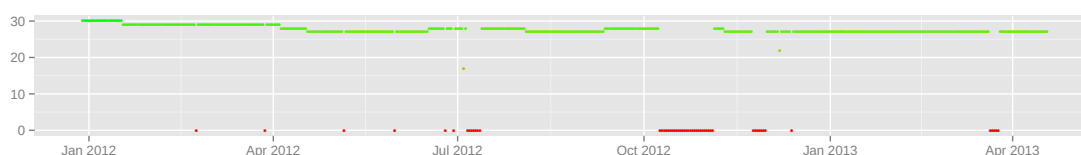


Figure 5.1: Number of files retrieved per day.

Image dimensions fluctuated daily—Table 5.1 and Table 5.2 contain low level information about image dimensions on the collection, displaying, respectively, width and height size properties, in pixels. With this information we can assert that *Público* and *Destak* were the ones that had the lowest image size variation and *Correio da Manhã* and *Metro - Grande Lisboa* had the highest variation.

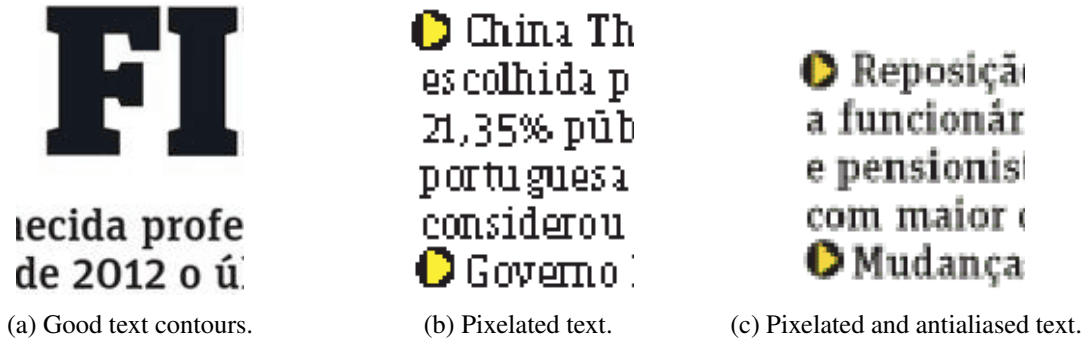
Also related with low level information, Table 5.3 displays the image size properties, in kilobytes, from the complete collection and for selected publications. Here, *Destak* and *Jornal de Notícias* were the most constant, and *Correio da Manhã* and *Jornal i* were the most inconstant in terms of file size.

Table 5.1: Collection and selected publications width properties, in pixels.

PUBLICATION	MEAN	STD. DEV.	MIN	MAX
All Publications	1185.64	349.65	250	1673
Jornal de Notícias	1482.35	155.41	466	1590
Correio da Manhã	1375.95	314.62	414	1598
Público	1517.57	63.43	626	1673
Diário de Notícias	1404.38	145.23	420	1611
Jornal i	1318.27	108.97	1067	1513
Destak	1208.68	97.05	701	1646
Metro - Grande Lisboa	1322	223.16	600	1613

Image size is an important factor in order to correctly to extract the text content from the front page image. According to the *Tesseract* FAQ [46] page, text with an x-height below 10 pixels will have a low chance of accurate recognition. Although not a recurring problem, sometimes images from the collection are small, e.g. around 450x600 pixels, and it is indeed very difficult to correctly extract text characters from them, not only due to the image size, but also because of the noise adjacent to the characters, possibly from downsampling operations on the image.

Similarly related to noise, and even though most newspapers had sharp text edges, some of Metro’s front pages were pixelated at 100% zoom as Figure 5.2 shows. Despite this the *Tesseract* library was able to detect, to some extent, the available text.

Figure 5.2: Samples of irregular text quality from the *Metro - Grande Lisboa* newspaper.

5.2 Information Extraction

To extract information from this collection of front page images a *Ruby* script was created, which iterated through all the image files and processed them.

This processing means creating a SHA-1 hash for the file, checking if that file was already processed, i.e. a repeated front page, if it was not previously processed then we extract the keywords from the front page and store all that information (hash, date, publication and content) in a

Table 5.2: Collection and selected publications height properties.

PUBLICATION	MEAN	STD. DEV.	MIN	MAX
All Publications	1574.58	449.98	334	2000
Jornal de Notícias	1957.57	205.13	601	2000
Correio da Manhã	1797.74	407.55	529	2000
Público	1866.49	94.73	757	2000
Diário de Notícias	1849.15	196.82	561	2000
Jornal i	1846.80	146.39	1483	2000
Destak	1499.89	120.29	856	2000
Metro - Grande Lisboa	1679.51	276.57	808	1999

persisted object. As a side note, since the processed information is persisted in file, it is possible to stop the execution of the collection analysis and resume later on from the same point.

Once all file are processed, the script iterates through the persisted file’s content and generates an Extensible Markup Language (XML) file with that information using a structure that is compatible with Apache Solr. This XML file is used to create the document index on Solr. Figure 5.3 shows a visual overview of this process.

Some simple experiments were conducted using the `facedetect` example [43], provided in the OpenCV default install, to test the feasibility of extracting extra information from the newspaper images. Promising results were attained—Figure 5.4 shows the results of these experiments.

5.3 Conclusion

Since the collection is composed from all the available files for a given day in the *Banca de Jornais SAPO* newspaper archive, it has repeated covers for weekly, bi-weekly and monthly publications.

After analysis we asserted that there were available 3,842 unique images, representing single issues from 36 different publications. Table 5.4 shows the issues distribution per publication and Figure 5.5 represents the publications lifespan as they appear on our collection.

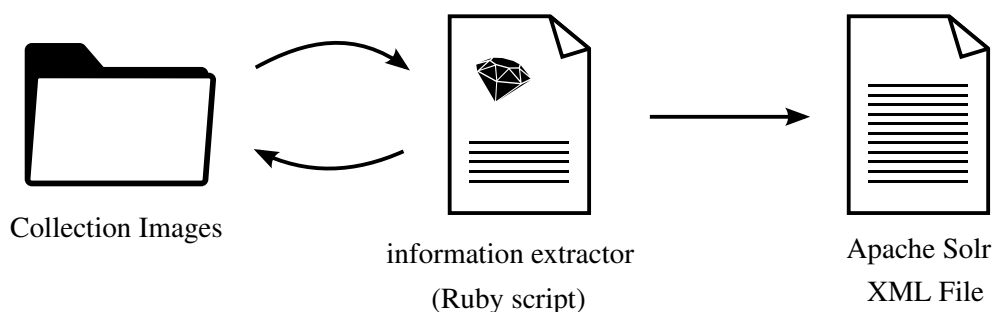


Figure 5.3: Collection information extraction.

Table 5.3: Collection and selected publications size properties, in kilobytes.

PUBLICATION	MEAN	STD. DEV.	MIN	MAX
All Publications	613.34	366.12	34.50	2859
Jornal de Notícias	655.42	141.36	131	1443.84
Correio da Manhã	668.67	218.05	115	1093.63
Público	688.39	188.16	170	1486.85
Diário de Notícias	640.68	168.38	126	1228.80
Jornal i	504.87	211.70	133	1250.30
Destak	408.14	94.80	136	828
Metro - Grande Lisboa	479.72	168.06	167	1049.60

Further analysis, using the algorithm described in Chapter 6, reveals the distinct layouts that were present on the various front pages. Table 5.5 contains the number of rectangular regions detected on different publications. Here, *Correio da Manhã*, *Jornal de Notícias* and *Destak* had the highest number of rectangular regions, this meaning that their layouts clearly separate individual content using rectangles and coloured regions, while the other publications had a lower rate of regions, meaning that the content is separated using gutters and line separators instead of rectangles.

Newspaper collection



Figure 5.4: Face detection experiments using the OpenCV library.

Table 5.4: Publication issues count.

PUBLICATION	ISSUES	PUBLICATION	ISSUES
Jornal de Notícias	422	P2 Público	50
Correio da Manhã	419	Sol Tabu	49
Público	419	Inimigo Público	48
Diário de Notícias	417	Domingo CM	41
Jornal i	367	Jornal de Letras, Artes e Ideias	39
Destak	260	Dinheiro Vivo	18
O Primeiro de Janeiro	258	Courrier International	17
Metro - Grande Lisboa	253	Le Monde Diplomatique	17
Sábado	81	W Magazine	17
O Diabo	71	Seleções do Readers Digest	16
Avante	70	TimeOut Lisboa	16
Visão	70	Jornal Povo de Portugal	11
Expresso	67	Super Interessante	6
O Emigrante Mundo Português	65	National Geographic	5
Sol	65	Quero Saber	5
O Crime	64	TimeOut Porto	5
Notícias Magazine	59	Focus	3
Expresso Revista	51	5ª Dimensão	1

Newspaper collection

2011		2012												2013			
Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	
422 issues Jornal de Notícias																	
419 Correio da Manhã																	
419 Público																	
417 Diário de Notícias																	
367 Jornal i																	
260 Destak																	
258 O Primeiro de Janeiro																	
253 Metro - Grande Lisboa																	
81 Sábado																	
71 O Diabo																	
70 Avante																	
70 Visão																	
67 Expresso																	
65 O Emigrante Mundo Português																	
65 Sol																	
64 O Crime																	
59 Notícias Magazine																	
51 Expresso Revista																	
50 P2 Público																	
49 Sol Tabu																	
48 Inimigo Público																	
41 Domingo CM																	
39 Jornal de Letras, Artes e Ideias																	
18 Dinheiro Vivo																	
17 Courrier International																	
17 Le Monde Diplomatique																	
17 W Magazine																	
16 Seleções do Readers Digest																	
16 TimeOut Lisboa																	
11 Jornal Povo de Portugal																	
6 Super Interessante																	
5 National Geographic																	
5 Quero Saber																	
5 TimeOut Porto																	
1 Focus																	
1 5ª Dimensão																	

Figure 5.5: Publications issue count along time.

Newspaper collection

Table 5.5: Collection and selected publications rectangular regions count.

PUBLICATION	MEAN	STD. DEV.	MIN	MAX
All Publications	2.45	2.4	0	13
Jornal de Notícias	4.64	1.86	0	13
Correio da Manhã	6.23	2.91	0	13
Público	1.71	1.06	0	5
Diário de Notícias	1.19	0.97	0	6
Jornal i	1.63	0.97	0	5
Destak	4.15	1.68	0	7
Metro - Grande Lisboa	1.41	1.11	0	5

Newspaper collection

Chapter 6

newspaper-extractor: a keyword extractor for newspaper front pages

One of the main objectives of this work was the development of a keyword extraction method that targeted, specifically, newspaper front pages. This Chapter describes that method in greater detail and presents the results that were obtained.

6.1 Purpose

The purpose of this keyword extractor is to divide the newspaper front pages into manageable parts, in order to more accurately extract the available keywords, using a standard OCR library.

While initially it was intended that this method would divide the front page into news items and then process them, that turned out to be harder than expected, and the focus was shifted to improve the overall result accuracy when extracting keywords from the entire front page. This meant an increased focus in finding smaller regions with content from the front page, dividing the original problem into more manageable sub-problems, and also increased attention to the binarization method used to extract the keywords.

6.2 Algorithm

The *newspaper-extractor* was developed in *C++* and used the *OpenCV 2.3.1* [44] library to process the images, and the *Tesseract 3.0.2* [45] library to perform character recognition. The *gflags 2.0* [18] library was also used to easily allow command line flags to alter the behaviour of the program.

As an overview, depicted in Figure 6.1, the application receives an image and outputs the image's text content to a text file, as well as creating auxiliary debug files to a folder.

Using the command line flags, as was mentioned, it's possible to alter the behaviour of the extractor, e.g. choosing the binarization method to use, or even bypassing the extractor completely and using only the *Tesseract* library on the image—useful for quick comparison between extractor variations.

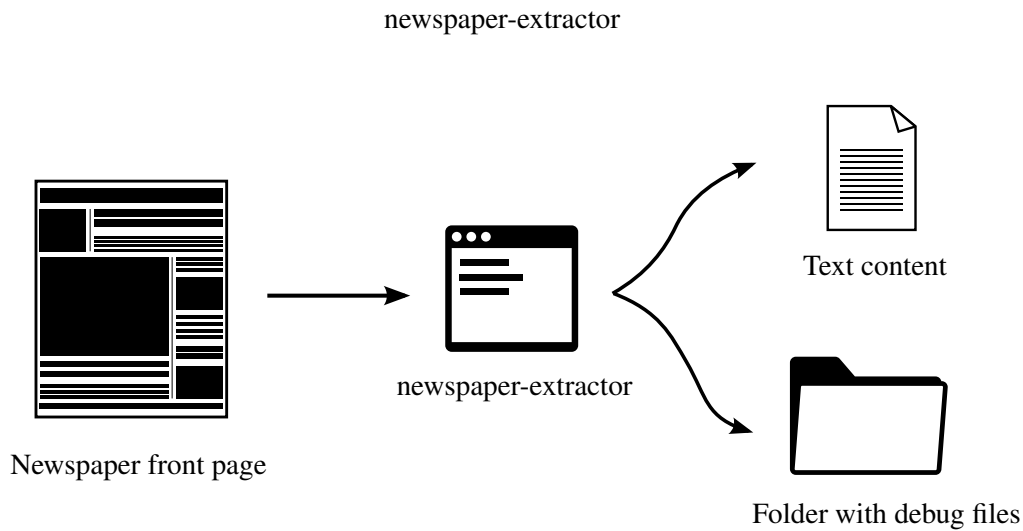


Figure 6.1: Artefact flow for the newspaper-extractor application.

The application starts by detecting the rectangular regions that are present in the image. This is done by finding only the extreme outer contours of the images, which are most external regions separated from other regions through a white gutter, i.e. the newspaper background colour. Figure 6.2 displays a newspaper front page and its outer contours.



Figure 6.2: Front page external contours.

Once the external regions are identified, they are filtered to select only the rectangular shaped ones. This filtering is a very basic one, the bounding box of the contour is obtained and if the width multiplied by the height, of that box, is approximate to the area of the contour then the region is kept, as it is a rectangular external region.

Also in this step, a first pass with the *Tesseract* library is done and rectangles with less than two detected text characters are discarded from the external rectangular list of regions. This is done because the rectangular regions that we are after are the ones surrounding news item, and some of the detected rectangles correspond only to single characters that have rectangular shapes. These discarded rectangles are, for example, the ones corresponding to capital letter “I” in sans-serif fonts or the capital “D”s whose shape is, in some fonts, similar to a rectangle. Figure 6.3 shows the detected rectangular regions, as well as false positives that were subsequently removed.

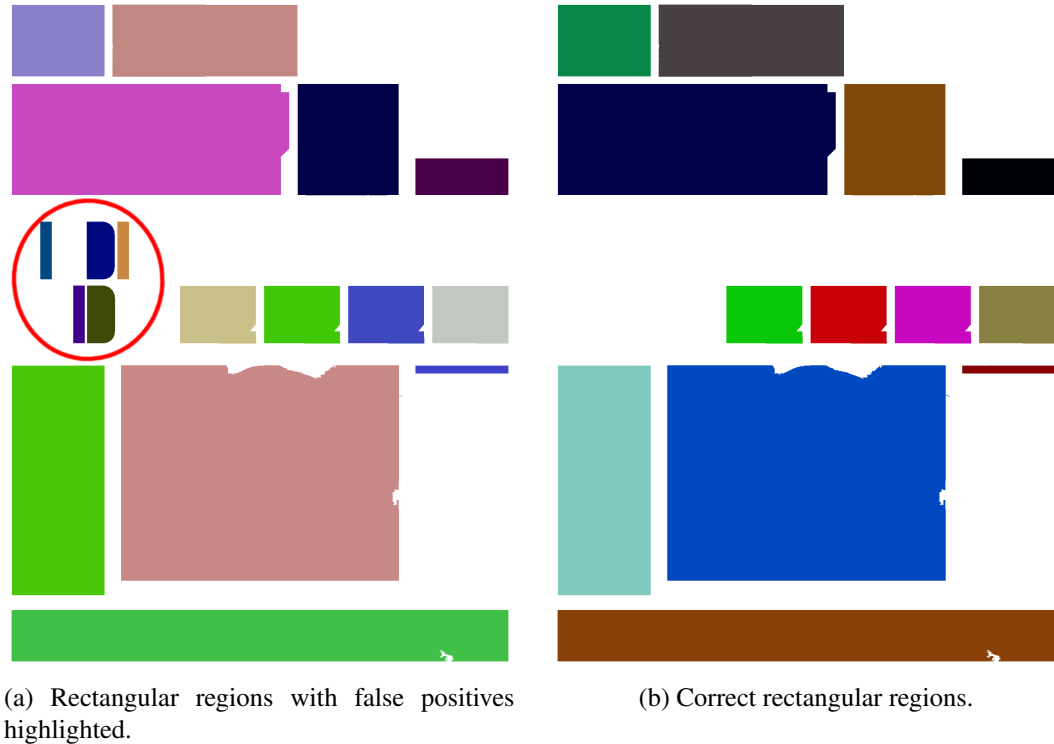


Figure 6.3: Detecting external rectangular regions from a front page.

The next step is to apply a binarization method to each of the detected rectangular regions. The default one is the Font and Background Color Independent (FBCI) text binarization method [24], but a global binarization method was also developed and can be used by passing a flag in the command line. The FBCI text binarization method has a different approach to binarization where the values of the edges from characters in the image are used to generate the correct threshold value for that particular edge. Also, this method always outputs a image where the foreground text colour as black and the background as white—this avoids further processing as that is the correct way to output text images to the *Tesseract* library.

Our implementation of the FBCI starts by using the Canny edge detection algorithm on each of the three colour channels of the input image. The thresholds used during this step are 0.66 and 1.33 times the mean channel value [50]. Then, the three resulting images are combined into one using a logical *OR* operation and the contours of that output image are archived. These contours correspond now to the text characters, pictures and dividing lines of the image, but since we want

only the text characters a quick filtering must be done first. Here we get the bounding box of the contour and keep only the ones where the aspect ratio, i.e. the width divided by the height of the box, fits between 0.1 and 10, and the box area is less then 1/5th of the area from the input image [24].

Once the contours are filtered we start estimating the thresholds for each contour. Here we set the *foreground-value* as the average colour from the border pixels for each contour, and set the *background-value* as the median value of a sample of twelve background pixels, i.e. three pixels from each of the corners of the contour's bounding box.

Now, if the foreground-value is lower than the background value, i.e. darker foreground than the background, the threshold function uses a binary operation, Equation 6.1, where if a pixel in the area of that contour, *src*, is higher than the calculated foreground-value, multiplied by an adjusting coefficient, then that same pixel in the resulting image, *bin*, is set as white—or black otherwise.

$$bin(x,y) = \begin{cases} white & \text{if } src(x,y) > \text{foreground-value} \times c1 \\ black & \text{otherwise} \end{cases} \quad (6.1)$$

If the foreground-value is equal or higher than the background value, i.e. lighter foreground than background, the threshold function used is now the inverted binary operation, Equation 6.2, similar to the one described earlier. The adjusting coefficients are $c1 = 1.05$ and $c2 = 0.95$. Modifying these values allows us to adjust how much of the surrounding of a character is included in the binarized character.

$$bin(x,y) = \begin{cases} black & \text{if } src(x,y) > \text{foreground-value} \times c2 \\ white & \text{otherwise} \end{cases} \quad (6.2)$$

While applying the FBCI method to each contour, in a rectangular region, an image is constructed with each individually binarized characters. Once this is done we have a rectangular region that consists only of black text over white background. Figure 6.4 and Figure 6.5 show the original rectangular region and the corresponding binary result using the FBCI text binarization method.

This binary region is then forwarded to the *Tesseract* library, with the language set to Portuguese, in order to extract the included text. This extracted text is stripped of any leading and trailing whitespace and is appended to the rest of the newspaper content.

Since the rectangular regions do not contain all of the newspaper content, they are subtracted from the full front page image and the result is then passed through the *Tesseract* library and its text content is also appended to the newspaper content. Figure 6.6 displays the remaining news items on the front page, i.e. the subtraction of the front page with the detected rectangular regions, and a composite with this remaining image and the binarized items that were obtained earlier.

This concludes the newspaper-extractor default process.



Figure 6.4: Binarization example with FBCI.

As an alternative to the FBCI binarization method, a global thresholding method was also developed. This starts by creating a histogram from the greyscale input image, where the number of occurrences for each colour are counted and stored. Now, from the histogram, the colour with the highest number of occurrences is assumed to be the background colour, and the colour with the second highest number of occurrences is assumed to be the foreground colour.

The thresholding is then performed similarly to the FBCI one described earlier, following the same logic and the same binary operations but using the colour with the highest number of occurrences as the *background-value* and the second highest as the *foreground-value*. This method, however, fails when a region has 3 or more distinct colours, with some content disappearing from the resulting binary image. Using Figure 6.4 and Figure 6.5 original regions it is possible to view in Figure 6.7 the global binarization thresholding in action, as well as the missing content.

6.3 Annotated Sample

With the focus shifted into improving the accuracy results of the extracted keywords, a strong test bed was essential to keep track of the small iterative improvements made to the newspaper-extractor. This test bed consists of an annotated sample of the collection and a *Ruby* script that runs the newspaper-extractor, with the different binarization methods, as well as a single pass on the front page using *Tesseract* and stores the results for further analysis.

The annotated sample consists of 35 front pages, 5 from each of the of daily newspaper publications: *Jornal de Notícias*, *Correio da Manhã*, *Público*, *Diário de Notícias*, *Jornal i*, *Destak* and *Metro - Grande Lisboa*. These were collected from January 15 through January 19 2012, however 3 of the sample front pages for *Jornal de Notícias*, from that timespan, had lower than average resolution. In order to keep the sample homogeneous, those 3 images were removed and 3 new *Jornal de Notícias* images, for January 22-24 2012, were added to the sample. Table A.1 and Table A.2, from Appendix A, contain low level information about the image dimensions from the

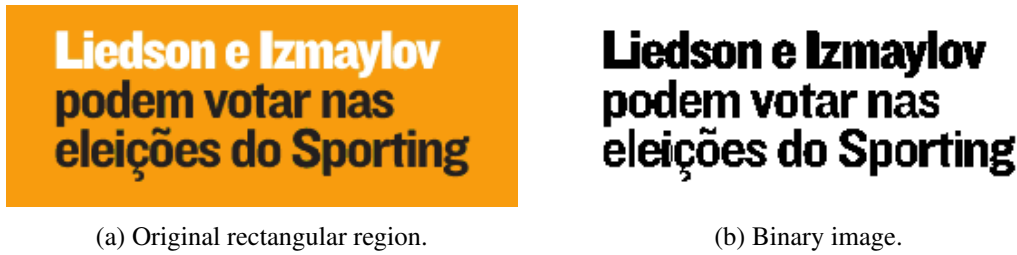


Figure 6.5: Another binarization example with FBCI.

sample. Comparing with the collection property analysis, *Destak* still maintains a low variation in image dimensions and *Correio da Manhã* still has the largest variation.

For each image an XML file was written with all of the visible text in the front page, encompassing the content of the news items, advertisements and miscellaneous text, e.g. names of directors and editors, publication name, weekday of the issue. An example for this sample annotation is displayed on Listing 6.1.

The structure of this file clearly separates the miscellaneous content from the publicity text, and keeps each news item in a separate section, distinguishing the title text from the body and extra text that could be present on the news item. Despite this, the comparator only uses the underlying text and ignores where it comes from, but future work could use this test bed to create and test a true news item segmenter, or test the correct detection of miscellaneous text or advertisement regions.

6.4 Results

To obtain results for the accuracy of the newspaper-extractor we created a *Ruby* script, which iterates through each of the image files in the sample directory and feeds each one into the newspaper-extractor multiple times, using different flags to control the binarization method to use and to do a pass using only *Tesseract* for comparison. To measure the accuracy of the method that is being tested, the script iterates through each of the words from the method's detected content file and checks if it is present on the sample list of words, if it is that word is removed from the sample and a correct match is counted. The accuracy value is the number of correct matches divided by the number of words on the sample. This is done for each of the images on the sample and for each test case, i.e. binarization with the FBCI method, binarization with the global method and using *Tesseract* on the input image. Figures A.1–A.7 display the resulting composite images per method variant, for a sample of the annotated newspapers.

Once all of the annotated sample images are processed the script then generates a comma-separated values (CSV) file with the mean accuracy value per publication, for each of the test cases. This CSV file can then be passed to an *R* [36] script which generates graphical visualisations of the results for quick visual comparison. Figure 6.8 and Figure 6.9 display this graphical visualisation, and the method variants present are:

```

1 <newspaper>
2   <item>
3     <title>Naufrágio</title>
4     <body>Mortos e desaparecidos numa "cena do Titanic" em Itália</body>
5     <pags>Pág. 30</pags>
6   </item>
7   <item>
8     <title>Música</title>
9     <body>Dançar, falar, criar: os melhores clubes do mundo, dos anos 20 ao século
10      XXI</body>
11     <pags>Pública</pags>
12   </item>
13   <misc>
14     Dom 15 Jan Edição Lisboa
15     Domingo, 15 de Janeiro de 2012, Ano XXII, nº 7951, 1,60€
16     Directora: Bárbara Reis
17     Directores adjuntos: Nuno Pacheco, Manuel Carvalho e Miguel Gaspar
18     Directora executiva Online: Simone Duarte; Directora de Arte: Sónia Matos
19     www.publico.pt
20   </misc>
21   <pub>
22     Amanhã Cadernos de Lanzarote
23     de José Saramago, Diário I
24     Nova colecção Amanhã, o primeiro de cinco volumes.
25     Por apenas mais € 5.50
26   </pub>
27   ...

```

Listing 6.1: Annotated sample example.



(a) Remaining news items on the front page.

(b) Binary rectangular regions added to the remaining items.

Figure 6.6: Remaining news items and resulting final output.

FBCI: This is the default newspaper-extractor, with FBCI binarization only on the rectangular regions.

FBCI on rem.: This is similar to the previous variation but also applies the FBCI binarization method on the remaining content after the rectangular regions removal.

Global: This uses the global binarization method described earlier instead of the FBCI. Does not binarize the remaining image.

Tesseract: This is the default test case. Applying only the *Tesseract* library on the input image.

The script also generates files with all the accuracy values for each test case—they are useful for the statistical analysis described next.

The *Tesseract* test case, i.e. using only *Tesseract* to extract the keywords, resulted in an average accuracy of 42.12%. While using the newspaper-extractor resulted in an average accuracy of 50.74%, an increase of 8.62 percentage points. Table 6.1 shows the accuracy result properties.

Using the generated files for each method variation, containing the accuracy values for each of the 35 sample front pages, it is possible to verify if the improvement of the newspaper-extractor, over the default *Tesseract* test case, is statistical significant. This was done using *R*'s one-sided paired t-test [28] to check if the mean accuracy of the newspaper-extractor sample results is greater than that of the *Tesseract* sample results. Through a *p-value* of 8.678×10^{-06} , we conclude that the mean accuracy of the newspaper-extractor sample is significantly higher than the mean accuracy

*“Dois anos seria
o ideal para
cumprir o déficit”*

Presidente
da C P

**podem votar nas
eleições do Sporting**

(a) Figure 6.4 region with global binarization.

(b) Figure 6.5 region with global binarization.

Figure 6.7: Global binarization method.

of the *Tesseract* sample. Despite the increase in number of correct words detected, there was also an increase in incorrect words detected using the new method—Table 6.2 shows this increase. Tables A.3–A.5, from Appendix A, contain more in-depth information of the results for each of the sample’s issues.

newspaper-extractor

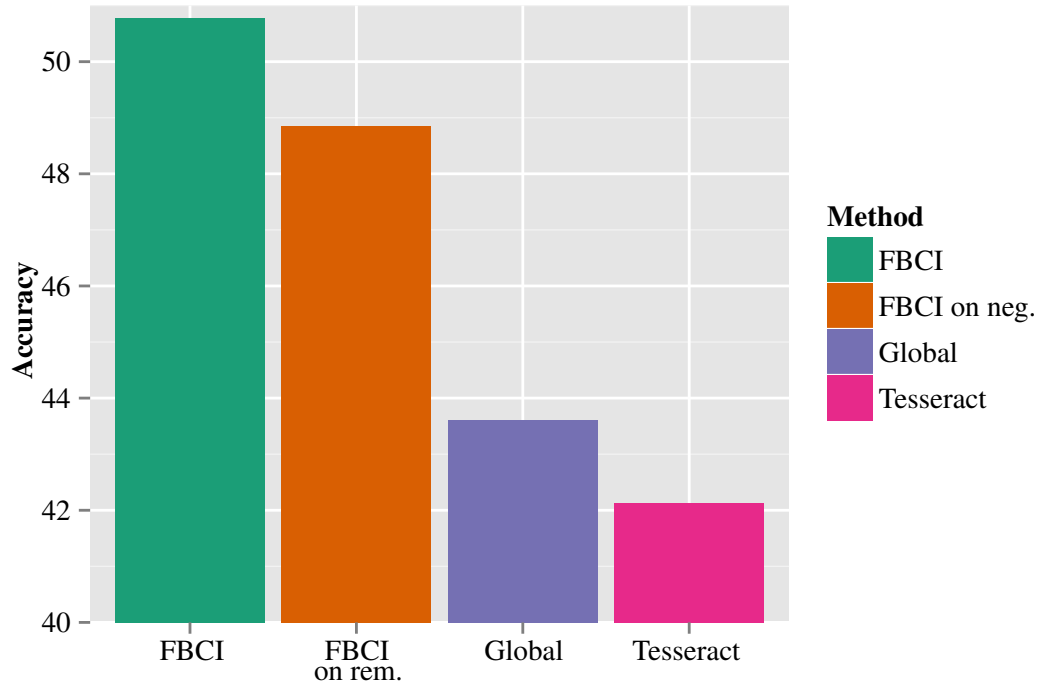


Figure 6.8: Comparison between method variants.

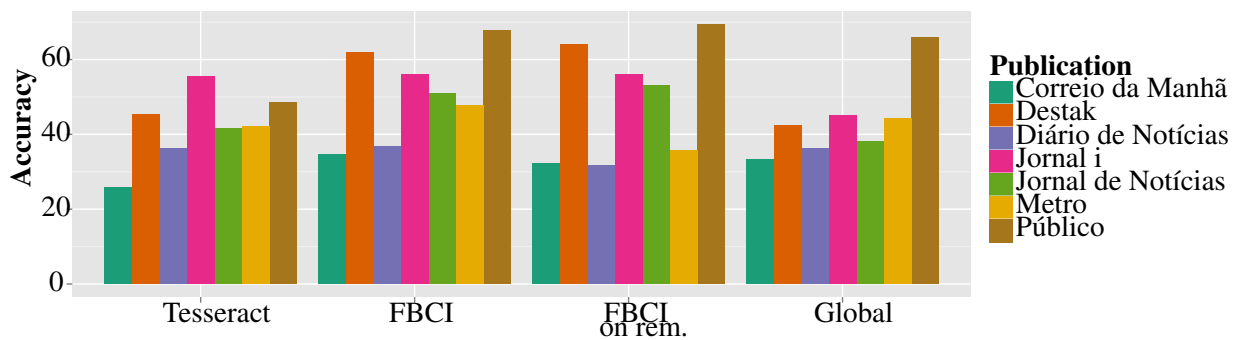


Figure 6.9: Method variants impact on the major publications.

Table 6.1: Accuracy properties per method variant.

METHOD VARIANT	MEAN	STD. DEV.	MIN	MAX
FBCI	50.74	13.38	24.91	77.11
FBCI on rem.	48.81	15.72	21.57	75.38
Global	43.62	14.68	9.94	77.59
Tesseract	42.12	11.54	18.69	67.19

Table 6.2: Number of incorrect words per method variant.

METHOD VARIANT	MEAN	STD. DEV.	MIN	MAX
FBCI	102.83	55.82	28	243
FBCI on rem.	108.51	52.42	42	216
Global	80.74	57.43	16	227
Tesseract	88.91	60.62	8	218

Chapter 7

Gazeta

The other main objective of this work was the development of a prototype application using the extracted information. A content-aware newspaper archive that allows the users to search for keywords, to view occurrences of the keywords along the time, and to browse issues that contain those keywords.

7.1 Purpose

This content-aware newspaper archive serves as a possible use for the information obtained from the application of the newspaper-extractor to the available collection. The prototype allows users to search for content that is in the newspaper front page, and to visualize the appearances of that content along time, i.e. seeing the time frames where that content had more relevance, as demonstrated in Figure 7.1, where occurrences for the keywords *Boston* and *maratona* appear only after April 15, 2013, the day of the *Boston Marathon bombings*.

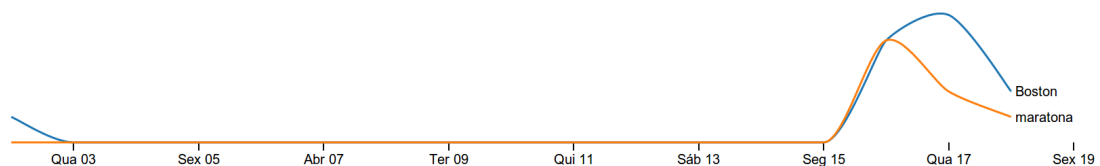


Figure 7.1: Occurrences for *Boston* and *maratona* for April 2–18, 2013.

One other visualization was created, but not further developed, which displays the top keywords over time. This provides a quick overview of what keywords were the most used for a specific timespan, displayed in Figure 7.2.

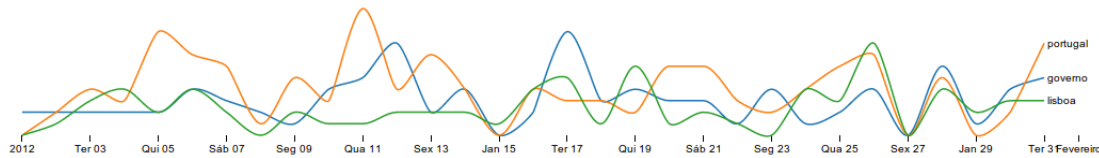


Figure 7.2: Keywords with the most number of occurrences for January, 2012.

7.2 Architecture

The system was built using Ruby on Rails [12], which provides a stable starting point for development. Apache Solr was used to store the information that was obtained from the collection extractor. Figure 7.3 gives an overview of the information flow in the prototype application.

As described in Chapter 5, the collection extractor script iterates through all the collection images and feeds them to the newspaper-extractor application in order to obtain its content. This script then generates an XML file with the documents, which are then inserted into the Solr index. This is represented as the dashed line in Figure 7.3.

In this index, each document is an issue from a publication, and contains the respective information: the date of the issue, the name of the publication, the corresponding image URL and the issue content as obtained through the newspaper-extractor. Keeping the information in the Apache Solr index allows us to use it as REST service, only sending requests to it when the user wants to know more information about a keyword. These requests contain a list of documents with their usual information and an extra field that shows the number of occurrences on that document for the requested keyword.

Listing 7.1 shows a request snippet for the keyword *Boston*, where each `doc` represents an issue and the `termfreq` field displays the number of occurrences for that keyword on that particular issue.

The result from the requests is then parsed on a Ruby model and adjusted to be displayed on the view. This means grouping and adding, from different issues, all occurrences for a particular day, and filling the days where there are no occurrences with 0 values, since those days do not appear on the Solr response and are necessary to correctly display the information on the view.

The views are built using Backbone.js [14], a JavaScript library that was used to enforce a coherent model structure, while creating the single page application. Although this is a relatively recent, and not completely mainstream technology, it was used to create a more interactive interface, so that browsing the archive would still be an interesting activity. The occurrences graph was created with D³ [5], a powerful and expansive JavaScript library that is used to visually display information.

The complete interface is shown in Figure 7.4. Here, the user searched for the keyword *saúde*. Looking at the resulting graph it is clear that there is a peak in keyword occurrences around April, 2012. Following are the steps that an user could take to obtain more information about this increase on the number of news about *saúde*.

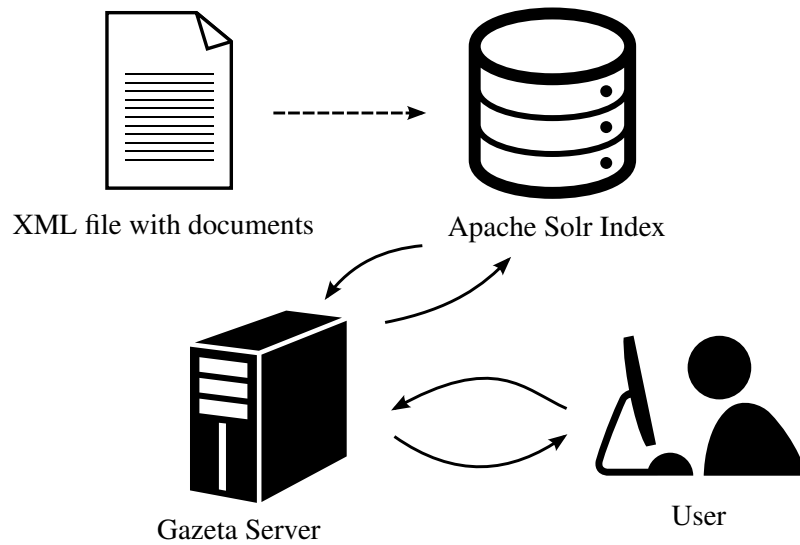


Figure 7.3: Request flow for the Gazeta prototype.

```

1 <response>
2   <result name="response" numFound="12" start="0">
3     <doc>
4       <str name="id">
5         d4760a34cba28be3b5b5e019d7eb630baf4699bf6de9f764ffb549ec...
6       </str>
7       <date name="date">2013-04-17T00:00:00Z</date>
8       <str name="url">2013-04-17/publico.png</str>
9       <str name="name">Público</str>
10      <int name="termfreq(content, "Boston")">3</int>
11    </doc>
12    <doc>
13      <str name="id">
14        e1654f35b3a85ab11891a7b4f48ccded34cd361f2f646f85fa7a1f7cb...
15      </str>
16      <date name="date">2013-04-16T00:00:00Z</date>
17      <str name="url">2013-04-16/metrolisboa.png</str>
18      <str name="name">Metro - Grande Lisboa</str>
19      <int name="termfreq(content, "Boston")">1</int>
20    </doc>
21    ...

```

Listing 7.1: Solr request response.

Figure 7.4: Occurrences for *saúde*.

The user could limit the search range by specifying a beginning and ending date. This would also limit the graph's range. As shown in Figure 7.5, this reduction in data points to display, smooths the graph's line and reveals that the peak occurred on April 12, 2012.

Clicking on some of the thumbnails, which were updated when user limited the date range, displays the corresponding newspaper front page in higher resolution, using a modal box. This allows the user to read headlines and content of that newspaper and understand why there was a peak in occurrences for *saúde* on that particular day. Accordingly to Figure 7.6 the Portuguese government was discussing a new law that would forbid adults from smoking inside cars when children are present.

With this information the user could now explore further, performing for example, a new search with *saúde, fumo* which would reveal that the keyword *fumo* also had an increase in occurrences for that specific day. This is displayed in Figure 7.7.

This previous sequence of Figures display most of the features present on the *Gazeta* newspaper archive. All data requests are done on the background, updating only the necessary parts of the interface, without a complete page refresh. The searched keywords are also added to the page URL after each request, replacing the previous one if present.

As seen in Figure 7.7, the application also allows the users to search for multiple keywords at once, by separating them with commas. This displays multiple lines on the graph, one for each keyword. One of the initial visualizations developed for *Gazeta* showed the keywords that had the highest number of occurrences for a specific timespan. This visualization, previously shown in Figure 7.2, reused the described comma separated query. However, if the requested timespan was too extensive the generated graph would be nearly unreadable. This is displayed in Figure 7.8.



Figure 7.5: Limiting the date range.



Figure 7.6: Newspaper issue detail.



Figure 7.7: Interface with multiple keyword search.

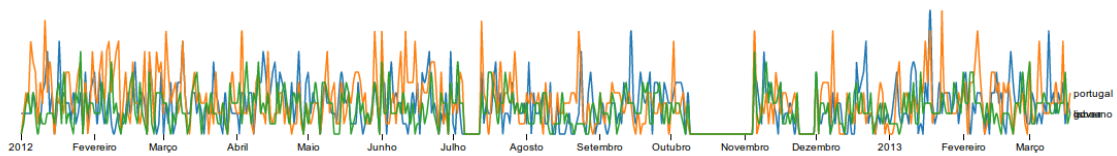


Figure 7.8: Top keywords for the entire newspaper collection.

Chapter 8

Conclusion

Although considered by some a dying medium, newspapers still have a great impact on the lives of many. The existence of a large number of different newspaper archives and aggregators is clear evidence that their information is still relevant to many people's daily routine. However, these newspaper archives sometimes have limited information, and limited search capabilities, to keep users interested in browsing beyond just the daily headlines.

Most segmentation methods used to extract information from newspapers and document images focuses on older newspapers and technical documents, dealing with stricter and simpler layouts without many images and colourful regions. While some of the available methods for page segmentation had positive and promising results, their implementation would be limited by the time and resource constraints of this study.

The decision of improving keyword extraction accuracy, specifically for current newspaper layouts, proved to be a simpler task and one with interesting results. The detection accuracy was improved by 8.62 percent points over using only the *Tesseract* library to extract the information, and this improvement was verified as having statistical significance using a one-sided paired t-test.

Extracting low level information from the available collection leads us to the conclusion that the newspapers with the lowest variation in image dimensions have the highest accuracy rating—*Público* and *Destak*. The newspaper with the highest variation also has the worst accuracy ratio of the selected publications—*Correio da Manhã*.

Developing a test bed from an annotated sample, despite being a tedious task, was very useful to keep track of the slight improvements on the application, and also to validate the benefits of a binarization method over another.

The development from scratch of the *Gazeta* content-aware newspaper archive was harder than expected and although it is useful as a prototype, significant changes would be needed in order to have a consumer-ready version. However, and despite the reduced number of features implemented, it is still possible to discover interesting news items, since it is easy to search for new content and to quickly browse the results.

8.1 Future Work

Along the development of this project many ideas sprouted but were cut short due to time and resource constraints. First, and one of the original ideas would be the creation of a true newspaper page segmenter instead of an improved keyword extractor. Separator line detection—a component that was actually developed but not used during this study—and gutter detection would be useful when detecting news items from a newspaper front page.

Using *template matching*, i.e. comparing all regions of a newspaper to another saved image, could be useful to detect miscellaneous regions on the front page, which should be removed from character extraction, to increase the relevance of the detected content.

The *Tesseract* API also provides some meta-information for the extracted text, e.g. font, size, boldness, and this font size information should be used to perform scaling operations that could improve a region's recognition accuracy.

Labelling or the correct categorization of the extracted keywords is also a possible improvement for the *Gazeta* newspaper archive. This would allow richer search components such as faceted search through the news categories.

Tests using real users should be done, in order to assess the usability of the *Gazeta* newspaper archive. The feedback obtained through these tests should be reviewed and if possible implemented to better immerse the user into the search interface.

References

- [1] A. Antonacopoulos, B. Gatos, and D. Bridson. ICDAR 2007 Page Segmentation Competition. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, volume 2, pages 1279 –1283, September 2007. doi: 10.1109/ICDAR.2007.4377121.
- [2] Apostolos Antonacopoulos, Stefan Pletschacher, David Bridson, and Christos Papadopoulos. ICDAR 2009 Page Segmentation Competition. *2009 10th International Conference on Document Analysis and Recognition*, pages 1370–1374, 2009. doi: 10.1109/ICDAR.2009.275.
- [3] Fundação ao Mário Soares. Diário de Lisboa. URL http://www.fmsoares.pt/diario_de_lisboa/ano. Accessed January 30, 2013.
- [4] Ori Ben-Yitzhak, Sivan Yogev, Nadav Golbandi, Nadav Har’El, Ronny Lempel, Andreas Neumann, Shila Ofek-Koifman, Dafna Sheinwald, Eugene Shekita, and Benjamin Sznajder. Beyond Basic Faceted Search. In *Proceedings of the international conference on Web search and web data mining - WSDM ’08*, page 33, New York, New York, USA, 2008. ACM Press. ISBN 9781595939279. doi: 10.1145/1341531.1341539.
- [5] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D³: Data-Driven Documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–9, December 2011. ISSN 1941-0506. doi: 10.1109/TVCG.2011.185.
- [6] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. O’Reilly Media, 2008. ISBN 978-0-596-51613-0.
- [7] Thomas M Breuel. Two Geometric Algorithms for Layout Analysis. In *Proceedings of the 5th International Workshop on Document Analysis Systems V, DAS ’02*, pages 188–199, London, UK, 2002. Springer-Verlag. ISBN 3-540-44068-2.
- [8] John Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, November 1986. ISSN 0162-8828. doi: 10.1109/TPAMI.1986.4767851.
- [9] Krishnendu Chaudhury, Ankur Jain, Sriram Thirthala, Vivek Sahasranaman, Shobhit Saxena, and Selvam Mahalingam. Google Newspaper Search – Image Processing and Analysis Pipeline. In *2009 10th International Conference on Document Analysis and Recognition*, pages 621–625. IEEE, 2009. ISBN 978-1-4244-4500-4. doi: 10.1109/ICDAR.2009.272.
- [10] Patrick Chiu, Francine Chen, and Laurent Denoue. Picture Detection in Document Page Images. In *Proceedings of the 10th ACM symposium on Document engineering - DocEng ’10*, page 211, New York, New York, USA, 2010. ACM Press. ISBN 9781450302319. doi: 10.1145/1860559.1860605.

REFERENCES

- [11] S. Chowdhury, S. Mandal, A. Das, and B. Chanda. Segmentation of Text and Graphics from Document Images. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007) Vol 2*, pages 619–623. IEEE, September 2007. ISBN 0-7695-2822-8. doi: 10.1109/ICDAR.2007.4376989.
- [12] Rails core team. Ruby on Rails. URL <http://rubyonrails.org/>. Accessed March 5, 2013.
- [13] Arganil City Council. A Comarca de Arganil. URL <http://acomarcadearganil.cm-arganil.pt/>. Accessed January 30, 2013.
- [14] DocumentCloud. Backbone.js. URL <http://backbonejs.org/>. Accessed March 5, 2013.
- [15] Apache Software Foundation. Apache Lucene - Apache Solr. URL <http://lucene.apache.org/solr/>. Accessed January 27, 2013.
- [16] Benjamin Jotham Fry. *Computational Information Design*. PhD thesis, Massachusetts Institute of Technology, 2004.
- [17] Basilios Gatos, Dimitrios Danatsas, Ioannis Pratikakis, and Stavros J Perantonis. Automatic Table Detection in Document Images. In *Proceedings of the Third international conference on Advances in Pattern Recognition - Volume Part I, ICAPR'05*, pages 609–618, Berlin, Heidelberg, 2005. Springer-Verlag. ISBN 3-540-28757-4, 978-3-540-28757-5. doi: 10.1007/11551188_67.
- [18] gflags Dev Team. gflags - Commandline flags module for C++ - Google Project Hosting. URL <https://code.google.com/p/gflags/>. Accessed June 5, 2013.
- [19] Google. Google News Archive Search. URL <http://news.google.com/newspapers>. Accessed January 30, 2013.
- [20] Greg Landgraf — American Libraries Magazine. Google Ends Newspaper Digitization Project, May 2011. URL <http://americanlibrariesmagazine.org/news/05242011/google-ends-newspaper-digitization-project>. Accessed February 4, 2013.
- [21] Sachin Grover, Kushal Arora, and Suman K. Mitra. Text Extraction from Document Images Using Edge Information. In *2009 Annual IEEE India Conference*, pages 1–4. IEEE, 2009. ISBN 978-1-4244-4858-6. doi: 10.1109/INDCON.2009.5409409.
- [22] Marti A Hearst. *Search User Interfaces*. Cambridge University Press, 2009. ISBN 0521113792.
- [23] Anil K. Jain and Bin Yu. Document Representation and Its Application to Page Decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):294–308, March 1998. ISSN 01628828. doi: 10.1109/34.667886.
- [24] T Kasar, J Kumar, and AG Ramakrishnan. Font and Background Color Independent Text Binarization. *Second international workshop on camera-based document analysis and recognition*, pages 3–9, 2007. URL <http://www.imlab.jp/cbdar2007/proceedings/papers/01-1.pdf>.

REFERENCES

- [25] Kiosko. Today's newspapers. Today's press covers. Kiosko.net, 2008. URL <http://en.kiosko.net>. Accessed January 30, 2013.
- [26] Ming Lei, Jianyong Wang, Baojue Chen, and Xiaoming Li. Improved relevance ranking in WebGather. *Journal of Computer Science and Technology*, 16(5):410–417, September 2001. ISSN 1000-9000. doi: 10.1007/BF02948958.
- [27] Michael S. Lew, Nicu Sebe, Chabane Djeraba, and Ramesh Jain. Content-based multimedia information retrieval: State of the art and challenges. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2(1):1–19, February 2006. ISSN 15516857. doi: 10.1145/1126004.1126005.
- [28] Todos Logos. Paired Student's t-test, July 2009. URL <http://www.r-bloggers.com/paired-students-t-test/>. Accessed June 5, 2013.
- [29] John Maeda. *The Laws of Simplicity (Design, Technology, Business, Life)*. The MIT Press, August 2006. ISBN 0262134721.
- [30] J. Malik. Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. ISSN 01628828. doi: 10.1109/34.868688.
- [31] Newseum. Newseum | Today's Front Pages | Gallery View, 2001. URL <http://www.newseum.org/todaysfrontpages/>. Accessed January 30, 2013.
- [32] NewspaperArchive. Explore Historical Newspaper Archives Online. URL <http://newspaperarchive.com/>. Accessed January 30, 2013.
- [33] Jakob Nielsen. Usability 101: Introduction to Usability, August 2003. URL <http://www.nngroup.com/articles/usability-101-introduction-to-usability/>. Last modified January 4, 2012.
- [34] University of Salford. PRIMa - Pattern Recognition & Image Analysis. URL <http://www.primaresearch.org/>. Accessed February 6, 2013.
- [35] Público and Lusa. Por dia vendem-se menos 29 mil jornais generalistas, June 2012. URL <http://www.publico.pt/media/noticia/por-dia-vendemse-menos-29-mil-jornais-generalistas-1552423>. Accessed January 9, 2013.
- [36] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013. URL <http://www.R-project.org>. Accessed June 5, 2013.
- [37] SAPO. Nacional - Banca de jornais - SAPO Notícias. URL <http://noticias.sapo.pt/banca/>. Accessed February 5, 2013.
- [38] SAPO. SAPO - Portugal Online!, 1997. URL <http://www.sapo.pt>. Accessed February 5, 2013.
- [39] Bill Scott and Theresa Neil. *Designing Web Interfaces: Principles and Patterns for Rich Interactions*. O'Reilly Media, 2009. ISBN 9780596516253.
- [40] Stamen. In The News / Vox Delicii. URL <http://stamen.com/projects/inthenews>. Accessed June 11, 2013.

REFERENCES

- [41] OpenCV Dev Team. Face Recognition with OpenCV — OpenCV 2.4.3 documentation, . URL http://docs.opencv.org/modules/contrib/doc/facerec/facerec_tutorial.html. Accessed January 31, 2013.
- [42] OpenCV Dev Team. Cascade Classifier — OpenCV 2.4.3 documentation, December 2012. URL http://docs.opencv.org/doc/tutorials/objdetect/cascade_classifier/cascade_classifier.html. Accessed January 30, 2013.
- [43] OpenCV Dev Team. OpenCV - FaceDetection - OpenCV DevZone, June 2012. URL <http://code.opencv.org/projects/opencv/wiki/FaceDetection>. Accessed January 30, 2013.
- [44] OpenCV Dev Team. OpenCV, August 2012. URL <http://opencv.org/>. Accessed February 4, 2013.
- [45] Tesseract Dev Team. tesseract-ocr - An OCR Engine that was developed at HP Labs between 1985 and 1995... and now at Google. - Google Project Hosting, . URL <https://code.google.com/p/tesseract-ocr/>. Accessed February 4, 2013.
- [46] Tesseract Dev Team. FAQ - tesseract-ocr - Frequently Asked Questions, . URL <http://code.google.com/p/tesseract-ocr/wiki/FAQ>. Accessed June 5, 2013.
- [47] Edward R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT, 2nd edition, 2001. ISBN 978-0961392147.
- [48] Marcos Weskamp. Newsmap, March 2004. URL <http://marumushi.com/projects/newsmap>. Accessed February 1, 2013.
- [49] InfoVis Wiki. "In The News" - Interactive Visualization of Google News, June 2005. URL [http://www.infovis-wiki.net/index.php?title=2005-07-21:_\"In_The_News\"_\"_Interactive_Visualization_of_Google_News](http://www.infovis-wiki.net/index.php?title=2005-07-21:_\). Accessed June 11, 2013.
- [50] Kerry Wong. Canny Edge Detection Auto Thresholding, May 2009. URL <http://www.kerrywong.com/2009/05/07/canny-edge-detection-auto-thresholding/>. Accessed March 20, 2013.
- [51] Ka-Ping Yee, Kirsten Swearingen, Kevin Li, and Marti Hearst. Faceted Metadata for Image Search and Browsing. *Proceedings of the conference on Human factors in computing systems - CHI '03*, page 401, 2003. doi: 10.1145/642679.642681.

Appendix A

Sample Results Appendix

A.1 Sample Properties

Table A.1: Sample publications width properties, in pixels.

PUBLICATION	MEAN	STD. DEV.	MIN	MAX
All Sample	1375.49	176.96	863	1568
Jornal de Notícias	1500	27.95	1460	1531
Correio da Manhã	1412.2	307.31	863	1568
Público	1522	21.99	1502	1547
Diário de Notícias	1273.8	1.64	1272	1276
Jornal i	1207.8	151.83	1067	1456
Destak	1215.2	0.45	1215	1216
Metro - Grande Lisboa	1497.4	27.73	1485	1547

Table A.2: Sample publications height properties, in pixels.

PUBLICATION	MEAN	STD. DEV.	MIN	MAX
All Sample	1800.2	219.38	1232	2000
Jornal de Notícias	1991.2	12.38	1974	2000
Correio da Manhã	1846.4	343.46	1232	2000
Público	2000	0	2000	2000
Diário de Notícias	1688	6.96	1676	1693
Jornal i	1700.8	176.34	1546	2000
Destak	1485.8	5.72	1478	1493
Metro - Grande Lisboa	1889.2	9.09	1882	1901

A.2 Sample Accuracy Results

Sample Results Appendix

Table A.3: Total number of words detected, for the issues, per method variant.

ISSUE	TOTAL	FBCI	FBCI ON REM.	GLOBAL	TESSERACT
Público - 2012-01-16	398	388	426	366	353
CM - 2012-01-16	231	213	210	165	163
Destak - 2012-01-16	224	207	208	139	117
JN - 2012-01-16	248	204	211	185	198
Jornal i - 2012-01-16	128	123	121	116	116
Metro - 2012-01-16	155	176	182	115	106
DN - 2012-01-16	631	389	363	362	436
Público - 2012-01-19	529	518	510	518	462
CM - 2012-01-19	289	170	168	113	154
Destak - 2012-01-19	278	192	224	111	163
Jornal i - 2012-01-19	161	197	201	32	122
Metro - 2012-01-19	158	159	156	106	150
DN - 2012-01-19	525	404	390	368	373
Público - 2012-01-17	294	239	233	239	198
CM - 2012-01-17	306	218	202	210	190
Destak - 2012-01-17	205	186	187	110	109
JN - 2012-01-17	209	175	171	121	132
Jornal i - 2012-01-17	142	98	113	98	98
Metro - 2012-01-17	209	228	224	149	221
DN - 2012-01-17	530	409	339	394	378
JN - 2012-01-23	276	248	231	175	231
JN - 2012-01-24	230	227	228	143	157
Público - 2012-01-15	215	176	224	164	100
CM - 2012-01-15	229	156	164	164	152
Destak - 2012-01-15	203	198	183	160	124
Jornal i - 2012-01-15	158	137	161	136	132
Metro - 2012-01-15	190	194	285	194	194
DN - 2012-01-15	360	327	325	313	360
JN - 2012-01-22	226	191	195	114	137
Público - 2012-01-18	415	427	417	426	285
CM - 2012-01-18	235	229	238	175	169
Destak - 2012-01-18	230	212	212	136	152
Jornal i - 2012-01-18	140	125	113	125	125
Metro - 2012-01-18	200	190	192	163	144
DN - 2012-01-18	500	429	368	422	348

Sample Results Appendix

Table A.4: Number of correct words, for each issue, per method variant.

ISSUE	TOTAL	FBCI	FBCI ON REM.	GLOBAL	TESSERACT
Público - 2012-01-16	398	245	300	216	191
CM - 2012-01-16	231	94	94	95	79
Destak - 2012-01-16	224	154	154	110	89
JN - 2012-01-16	248	125	141	111	101
Jornal i - 2012-01-16	128	82	79	76	86
Metro - 2012-01-16	155	65	56	62	53
DN - 2012-01-16	631	196	188	182	218
Público - 2012-01-19	529	325	366	325	292
CM - 2012-01-19	289	72	73	40	54
Destak - 2012-01-19	278	141	159	75	118
Jornal i - 2012-01-19	161	96	99	16	87
Metro - 2012-01-19	158	70	46	65	71
DN - 2012-01-19	525	201	174	201	203
Público - 2012-01-17	294	205	176	206	167
CM - 2012-01-17	306	105	68	121	59
Destak - 2012-01-17	205	135	132	89	86
JN - 2012-01-17	209	106	114	72	87
Jornal i - 2012-01-17	142	57	69	57	57
Metro - 2012-01-17	209	89	66	76	80
DN - 2012-01-17	530	166	138	167	164
JN - 2012-01-23	276	121	124	99	135
JN - 2012-01-24	230	133	124	103	80
Público - 2012-01-15	215	148	161	143	92
CM - 2012-01-15	229	69	76	90	47
Destak - 2012-01-15	203	116	124	106	103
Jornal i - 2012-01-15	158	92	95	92	92
Metro - 2012-01-15	190	119	87	119	119
DN - 2012-01-15	360	128	124	123	158
JN - 2012-01-22	226	116	123	68	94
Público - 2012-01-18	415	320	282	322	166
CM - 2012-01-18	235	100	94	77	84
Destak - 2012-01-18	230	153	157	93	118
Jornal i - 2012-01-18	140	81	67	81	81
Metro - 2012-01-18	200	94	71	83	62
DN - 2012-01-18	500	241	176	240	164

Sample Results Appendix

Table A.5: Number of incorrect words, for each issue, per method variant.

ISSUE	TOTAL	FBCI	FBCI ON REM.	GLOBAL	TESSERACT
Público - 2012-01-16	398	143	126	150	162
CM - 2012-01-16	231	119	116	70	84
Destak - 2012-01-16	224	53	54	29	28
JN - 2012-01-16	248	79	70	74	97
Jornal i - 2012-01-16	128	41	42	40	30
Metro - 2012-01-16	155	111	126	53	53
DN - 2012-01-16	631	193	175	180	218
Público - 2012-01-19	529	193	144	193	170
CM - 2012-01-19	289	98	95	73	100
Destak - 2012-01-19	278	51	65	36	45
Jornal i - 2012-01-19	161	101	102	16	35
Metro - 2012-01-19	158	89	110	41	79
DN - 2012-01-19	525	203	216	167	170
Público - 2012-01-17	294	34	57	33	31
CM - 2012-01-17	306	113	134	89	131
Destak - 2012-01-17	205	51	55	21	23
JN - 2012-01-17	209	69	57	49	45
Jornal i - 2012-01-17	142	41	44	41	41
Metro - 2012-01-17	209	139	158	73	141
DN - 2012-01-17	530	243	201	227	214
JN - 2012-01-23	276	127	107	76	96
JN - 2012-01-24	230	94	104	40	77
Público - 2012-01-15	215	28	63	21	8
CM - 2012-01-15	229	87	88	74	105
Destak - 2012-01-15	203	82	59	54	21
Jornal i - 2012-01-15	158	45	66	44	40
Metro - 2012-01-15	190	75	198	75	75
DN - 2012-01-15	360	199	201	190	202
JN - 2012-01-22	226	75	72	46	43
Público - 2012-01-18	415	107	135	104	119
CM - 2012-01-18	235	129	144	98	85
Destak - 2012-01-18	230	59	55	43	34
Jornal i - 2012-01-18	140	44	46	44	44
Metro - 2012-01-18	200	96	121	80	82
DN - 2012-01-18	500	188	192	182	184

A.3 Binarization Visual Comparison

Sample Results Appendix



(a) Original Newspaper.

(b) Default newspaper-extractor binarization.



(c) Binarization with FBCI method and also on the remaining.



(d) Binarization with global method.

Figure A.1: *Correio da Manhã* different binarization methods.

Sample Results Appendix



Figure A.2: *Destak* different binarization methods.



(a) Original Newspaper.



(b) Default newspaper-extractor binarization.



(c) Binarization with FBCI method and also on the remaining.



(d) Binarization with global method.

Figure A.3: *Diário de Notícias* different binarization methods.

Sample Results Appendix



(a) Original Newspaper.



(b) Default newspaper-extractor binarization.



(c) Binarization with FBCI method and also on the remaining.



(d) Binarization with global method.

Figure A.4: *Jornal i* different binarization methods.



(a) Original Newspaper.



(b) Default newspaper-extractor binarization.



(c) Binarization with FBCI method and also on the remaining.



(d) Binarization with global method.

Figure A.5: Metro - Grande Lisboa different binarization methods.

Sample Results Appendix



(a) Original Newspaper.



(b) Default newspaper-extractor binarization.



(c) Binarization with FBCI method and also on the remaining.



(d) Binarization with global method.

Figure A.6: *Público* different binarization methods.

Tinham seis escravos e recebiam Rendimento Mínimo

Valongo A sogra Caminda, o genro Mário Fernando e o capataz Ludovino escravizaram ucraniano, deficiente sem dedos, epilético, sem-abrigo, demente e inválido, andavam de Mercedes e tinham contas bancárias em Espanha [10]



(a) Original Newspaper.

Tinham seis escravos e recebiam Rendimento Mínimo

Valongo A sogra Caminda, o genro Mário Fernando e o capataz Ludovino escravizaram ucraniano, deficiente sem dedos, epilético, sem-abrigo, demente e inválido, andavam de Mercedes e tinham contas bancárias em Espanha [10]



(b) Default newspaper-extractor binarization.

Tinham seis escravos e recebiam Rendimento Mínimo

Valongo A sogra Caminda, o genro Mário Fernando e o capataz Ludovino escravizaram ucraniano, deficiente sem dedos, epilético, sem-abrigo, demente e inválido, andavam de Mercedes e tinham contas bancárias em Espanha [10]



(c) Binarization with FBCI method and also on the remaining.

Tinham seis escravos e recebiam Rendimento Mínimo

Valongo A sogra Caminda, o genro Mário Fernando e o capataz Ludovino escravizaram ucraniano, deficiente sem dedos, epilético, sem-abrigo, demente e inválido, andavam de Mercedes e tinham contas bancárias em Espanha [10]



(d) Binarization with global method.

Figure A.7: *Jornal de Notícias* different binarization methods.

A.4 Complete annotated sample.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <newspaper>
3   <item>
4     <title>JOSÉ MAGALHÃES</title>
5     <body>Remodelação de gabinete custa 62 mil €</body>
6     <pags>PÁGS.10 E 11</pags>
7   </item>
8   <item>
9     <title>'Titanic' em Itália fere portugueses</title>
10    <body>Paquete de luxo embate em rochas e naufraga. Operação caótica de
        salvamento com 3 mortos e 40 desaparecidos.</body>
11    <pags>PÁGS.4 E 5</pags>
12  </item>
13  <item>
14    <title>FÉRIAS PAGAS A MILITARES</title>
15    <extra>500 RECEBEM SUBSÍDIO ESTE MÊS</extra>
16    <body>Ministério da Defesa considera o pagamento a reservistas das Forças
        Armadas</body>
17    <pags>PÁG.33</pags>
18  </item>
19  <item>
20    <extra>BENFICA 4 | 1 V. SETÚBAL</extra>
21    <title>Goleada cimenta liderança da águia</title>
22    <body>Cardoso bisa e encarnados fecham primeira volta da Liga isolados</body>
23  </item>
24  <item>
25    <title>Hulk sai lesionado durante o jogo FC Porto-Rio Ave (2-0)</title>
26    <pags>PÁGS.6 A 9</pags>
27  </item>
28  <item>
29    <extra>ALGARVE</extra>
30    <title>Morto à pancada e queimado</title>
31    <pags>PÁG.12</pags>
32  </item>
33  <item>
34    <extra>GRÁVIDA</extra>
35    <title>Violada paga processo</title>
36    <pags>PÁG.20</pags>
37  </item>
38  <item>
39    <title>Sócrates almoça com Alberto Costa</title>
40    <body>Em restaurante de Lisboa</body>
41    <pags>PÁGS.49 a 53</pags>
42  </item>
43  <item>
44    <title>Margarida Marinho confirma divórcio</title>

```

Sample Results Appendix

```
45 </item>
46 <pub>
47     SUPER CONCURSO
48     HOJE
49     GANHE
50     BARRAS DE OURO
51     COM O CM
52     ENCHA CARRINHO TODO O ANO
53     Prémios contra a crise
54     13 600 EUROS EM VALES DE COMPRAS POR SEMANA
55     PÁG. 34
56     CONCORRA!
57     CUPÃO NA PÁGINA 55
58 </pub>
59 <pub>
60     Arlloufil
61     OURO PRATAS JÓIAS MOEDAS
62     Avaliamos e Compramos!
63     AVALIADOR OFICIAL
64     Casa da Moeda
65     50
66     Tel. Grátis: 800 208 779
67 </pub>
68 <pub>
69     NO SEU JORNAL
70     COMO PENSAM OS CHINESES QUE TÊM PODER NA ECONOMIA PORTUGUESA
71 </pub>
72 <pub>
73     ENQUANTO LÊS ESTE JORNAL ALGUÉM ESTÁ A RASPAR E A GANHAR
74     raspadinha
75     Está sempre a dar
76 </pub>
77 <misc>
78     DOMINGO 15/01/2012 € 1.30 (C/IVA)
79     CORREIO da manhã
80     www.cmjornal.pt
81     Director Octávio Ribeiro | Dir. Adj. Armando Esteves Pereira e Eduardo Dâmaso |
82     Subdir. Manuel Catarino
83 </misc>
84 </newspaper>
```

Listing A.1: Annotated sample for Figure A.1