1
Introduction

We face an ever growing amount of content being produced every day. In this scenario, identifying and extracting the contents of a webpage, discarding templates and similar non-relevant parts of the page, is useful for several applications. To name a few, screen reading software for the visually impaired may focus on the content and skip templates and other irrelevant content; search engines can more cleanly store the page's data, which provides more accurate search results; and small screen devices, such as modern mobile phones, can use it to increase readability.

The task of news segmentation consists in identifying the key regions of the webpage. These regions might have a smaller or bigger role depending on the application. For our purposes, we consider the title, publication date and story body as regions of interest. We illustrate the desired regions in Figure 1.1.

Very good results have already been reported for this task: using the F-score metric, which we describe in Section 2.5, we observe works reaching 97% of F1 for title detection [33, 35], 87% of F1 for publication date detection [35] and 94% of F1 for news body detection [35] for an exact DOM node metric. We believe that these results are satisfactory for some applications and there's no room for large improvements other than refining current approaches. However, most approaches rely on rendering the webpage, which demands lots of processing and, consequently, time.

Our focus is on large-scale document processing, specifically the page processing step of search engine's web crawlers, extending the previous work of [18]. Search engines keep local versions of the webpages for indexing purposes. With a cleaner copy of a page, search results tend to be much more relevant as terms that would otherwise be part of the page, but not part of the relevant content (templates, advertisements, etc.), are discarded. In addition, the knowledge of the news title and its publication date can be useful for ranking results, either by relevance (since the title is a general summary of the text) or by date. Some works in the literature have shown how information extraction can improve document retrieval, such as [14] and [46].

For this specific scenario, we feel that there is a lack of suitable ap-

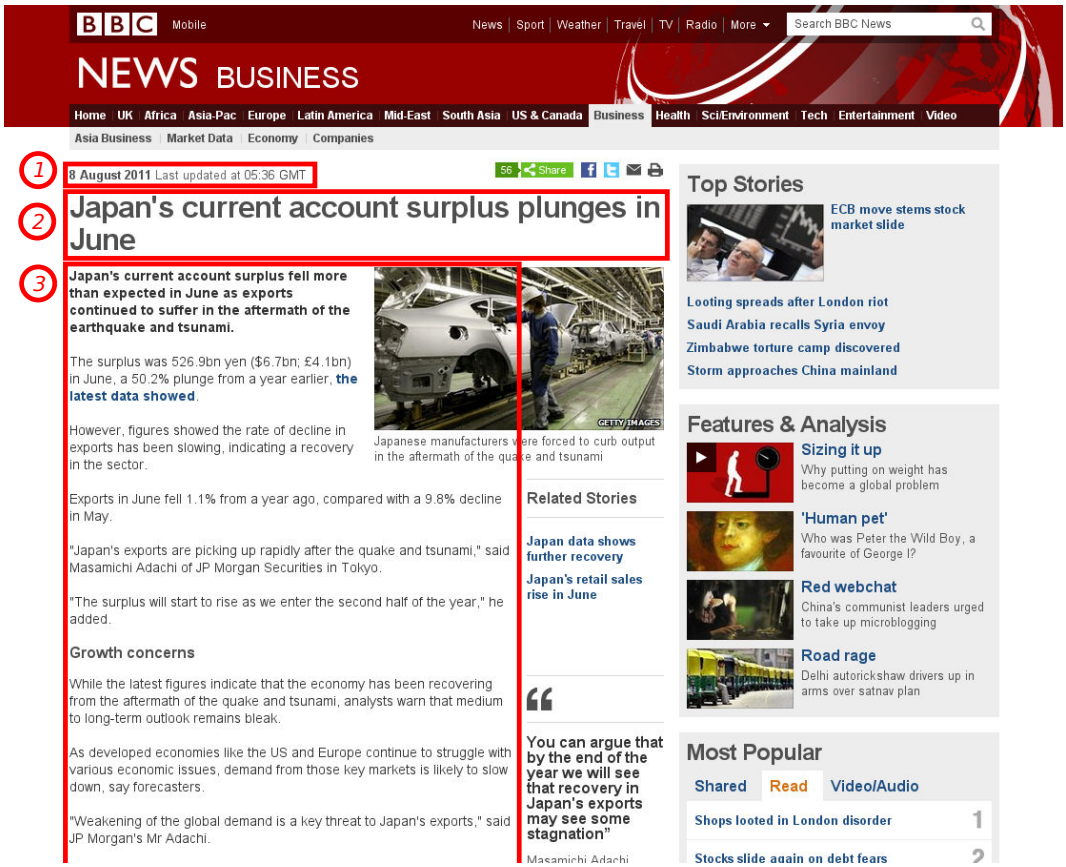

Figure 1.1: Illustration of the regions of interest in a news webpage. The rectangles identify the date, title and body, respectively numbered 1, 2 and 3.

proaches. Rendering webpages is not an option because it is a time-consuming task, as shown by [18, 22], and it would slow down the throughput of these systems. Thus, we propose a new approach that more closely keeps up with the high volume of documents in this scenario, while still producing satisfactory results.

1.1 Related work

Related work in the area may be classified in various ways. It is common to differentiate them by their scope, which creates the notions of *site-level* and *page-level approaches*, and by the requirements to solve the problem, which range from *strictly structural* properties to a *full rendering* of the page, which provides the geometric positioning of elements and allows the use of computer vision algorithms. Other possible forms of classification include the *language independence* of methods; that is, whether the language in which the news story was written impacts on the results.

Site-level approaches require some mass of example data to build a model or rules that are specific for its pages. As it is tailored for a specific group, the results are generally better, but come at the cost of high maintenance, high setup costs and limited usability due to the wrappers that are created to exploit particularities of each site's design. A good example of this approach is [39], which identifies site templates using tree edit distance.

On the opposite side of the spectrum lies the page-level approaches. These are devised to work on virtually any webpage, including those from websites never seen before. Its generality comes at the cost of slightly worse results, but requires little maintenance, has low setup costs and broad usability since the approach works independently of the site's design. Examples of these are [14, 46], which identify titles in generic webpages and [40, 41], which train a model from a single website and apply it to 11 others to extract the news article content.

The structural approaches depend on features directly extracted from the HTML file, which may or may not be converted into a DOM tree. Information such as number of nodes, link density, word tokens, among others, are considered structural features. Methods that make good use of them include [23], which describes one of the winning approaches used in the CleanEval shared task [5], and [28], which uses a token-based local classifier to identify the boundaries of article text.

The rendering approaches may include all features available structurally, but have access to other information such as geometric positioning, bounding box size, font size and font color of various elements in the webpage, commonly available in web browsers. Works making use of this information generally achieve better results than strictly structural ones, from which we may cite [33, 34, 35], which perform segmentation of news pages' content in several classes, and [22], which takes the approach of [28] and applies it to visual features with good results.

Some approaches are language-dependent, which means they explore the document's language to aid in the task. Possibilities include, for example, training a model that estimates the probability of a given word being part of the news story, as done in [28]. Other approaches, however, are language-independent, which means they can still perform well in documents written in any language. As a language-independent example we may cite [40, 41], which reports results in English-written pages but with good results on Chinese pages too.

1.2 Our contribution

Our contribution consists of an efficient language-independent approach for news segmentation. We avoid a full rendering of the webpage in favor of a partial rendering, which is key to keep the desired performance for high throughput systems.

First, our method locates a DOM node that includes the page's relevant content in its subtree. Then, we proceed to remove noise from this subtree. Finally, we use machine learning models that identify the title and publication date. For these models, we use structural features and visual presentation information computed by a simplified CSS parser. The reduced subset of the page in which we apply these models, along with the simplified CSS implementation, provides us with the necessary performance we were looking for.

To test our approach, we constructed a corpus consisting of 200 news webpages from 50 different domains. These pages are written in English, Portuguese and Spanish and have been manually annotated for news title, publication date and article text. We have carefully observed the often overlooked aspects of processing time and results quality when applying our model to a website not previously seen in training, as well as those already seen. Our approach has shown to be about an order of magnitude faster than an equivalent full rendering alternative while retaining a good quality of extraction. The results we obtained in a cross-validation for seen websites using this corpus, measured with a text-based metric, are 92% of F1 for title detection, 84% of F1 for date detection and 88% of F1 for body detection. For unseen websites, the results are 91% of F1 for title detection, 77% of F1 for date detection and 88% of F1 for body detection. Slightly better results are achieved when testing against other corpora such as the NEWS600 [34], which consists of 604 pages from 177 distinct domains, all written in English.

We also created another corpus, consisting of 10 sites with 300 pages each, with the intent of replicating results from two works in the literature. We use this corpus and these works to successfully validate our approach, with results on par with them: 88% of F1 for title detection, 72% of F1 for date detection and 88% of F1 for body detection on this new corpus, which we called Cardoso3000.

1.3 Organization

In Chapter 2 we will go over important concepts for the full understanding of this work. In Chapter 3 we will describe our approach to the task and in Chapter 4 we will discuss selected works from the literature and how we implemented them. In Chapter 5 we will go over our experiments for all approaches implemented (ours and the ones selected from the literature). In Chapter 6 we will present our closing thoughts and conclusions.