5
Experiments

In this chapter we will introduce the corpora used and go over some experiments we conducted for our approach and two other works in the literature that we implemented: Luo, et al. (2009) [22] and Wang, et al. (2009) [40, 41].

For the implemented works from the literature, we executed the experiments on a Linux platform, using the Python programming language [29] and the WebKit rendering engine [42] via the python-webkit library bindings [30]. We also used the Java programming language [16] and the JPype library [17] to intercommunicate Python code and Java, to facilitate the use of the Weka tool for machine-learning models [43] and the libxml2 [20] and libxml2dom [21] libraries to parse HTML for our simplified CSS parser. The machine used is a Intel Core 2 Duo @ 1.86 Ghz with 1 Gb of RAM running Ubuntu Linux 10.04 LTS 32-bit.

Experiments for our approach were executed on a Windows platform, using the C# programming language [3]. The machine used is a Intel Core 2 Quad @ 2.40 Ghz with 4 Gb of RAM running Windows Server 2008 64-bit.

5.1
Corpora

In this section we will go over our corpora – or *datasets* – of news webpages. We will start by describing the corpora available to us, then we will go over the annotation format and process used.

5.1.1
Available corpora

RCD4

We constructed the RCD4, a corpus of 200 news pages that we employed in our experiments. Its pages were manually obtained and manually annotated for news title, publication date and story body. The pages in the corpus are written in English, Portuguese and Spanish. All objects referred by the pages

are also downloaded, such as images, style sheets, script files, etc., in order to keep our copies as close as possible to the originals.

We obtained pages from 50 different domains, with 5 pages per domain being the most frequent sample, taking special care to organize these pages in such a way that allows us to perform cross-validation tests to evaluate how well our models generalize the problem for unseen domains and how results can be improved by adding examples of pages from the same domains. A thorough list of the domains in the corpus is available in Appendix A.

NEWS600

NEWS600 [34] is a publicly available corpus that consists of 604 pages from 177 distinct domains. All of its pages are written in English and were manually annotated with 37 labels, divided into six categories:

- Content: with information directly related to the news content. Contains the "Introduction" (an emphasized first paragraph), "Title", "Subtitle", "Category" (for instance, denoting a sports section), "Paragraph", "Heading" (some sort of subsections on the story) and "FurtherReading" labels;
- Creation: with metadata about the page, containing the "Date" (publication date), "Author", "AuthorPublisherDate" (for cases when author/publisher information cannot be separated from the publication date), "Place" (location or city where the story takes place), "Editor", "Publisher" (generally a news agency such as Reuters), "Contributor", "Contact" (contact information, such as e-mail, of author, publisher or editor) and "Copyright" labels;
- Navigation: regards pagination for long stories and contains the labels "CurrentPage", "PreviousPage", "NextPage", "FurtherPage" (direct links for further pages of the article, such as 3rd or 4th page, but not previous/next) and "LastPage";
- Media: for non-textual content directly related to the article. Contains the labels "Multimedia" (used for images, videos, infographs, etc.), "MediaCredit" and "Caption" labels;
- Advertisements: with the single "Advertisement" label; and
- Links: with other links that might be present in the document. Contains the "Topics", "Tags", "Comments" (only the link to comments, not the actual comment content), "Recommendation" (explicitly indicated as such), "EmailArticle", "SocialBookmark" (for instance, Twitter,

Facebook, Delicious, etc.), "RelatedArticle" (for instance, past coverage – not the same as the "FurtherReading" label), "SinglePage" (for viewing a multipage story in a single page), "PrintPage", "Rating" and "RSSFeed" labels.

Remapping of labels are also suggested for several applications. In our case, we did our own mapping of our labels of interest, as follows:

- Nodes annotated as "Title" are considered as title nodes.
- Nodes annotated as "Date" are considered as publication date.
- When no "Date" nodes are found in a document, nodes annotated as "AuthorPublisherDate" are considered as publication date.
- For this corpus, no body annotations were used.

The pages in this corpus were obtained in an attempt to mimic the general distribution of pages in the Internet. However, our personal evaluation is that very few pages from big news portals were gathered this way, which is an unrealistic representation of access given the habits of most users to share links by emails, instant messaging, social networks and/or websites such as Reddit[1]. Instead, we see a great amount of news posted in an organization's website and blog-like pages.

The domain with the highest number of pages was `www.bbc.co.uk` with 39 pages, followed by `www.canada.com` with 21 pages and `www.independent.ie` and `www.reuters.com`, both with 20 pages. A total of 119 domains have contributed with only one page, 36 domains contributed with two pages and 20 domains contributed with three pages. For other quantities of contributed pages, the total number of domains never exceeded 4.

Cardoso3000

This corpus consists of a total of 3000 pages from 10 different news portals, being 6 of them in English (BBC, CNN, FOXNEWS, NEWSWEEK, NYTimes and TIME) and 4 of them in Brazilian Portuguese (Estadão, Folha de São Paulo, G1 and Terra). From each portal, exactly 300 pages were obtained, along with all referenced files such as images, scripts, style sheets, etc., similarly to the other corpora described.

Differently from the previously described corpora, all pages in this corpus were obtained and annotated programatically. For each news portal, two scripts were written: the first one scraped the pages looking for news story URLs; and

[1] `http://www.reddit.com/`

the second one annotated title, date and body of the news story. This was necessary in order to obtain the amount of pages we wanted in a reasonable time frame.

This corpus was created specifically to benchmark approaches in the literature, since it is not uncommon to find results reported for these websites, and to validate our implementations and results.

In the process of writing the scripts for this corpus, we had to carefully analyze the structure of various news stories in a given news portal. This gave us some interesting insights on the work involved in writing custom wrappers and revealed some curiosities. Some websites have a more homogeneous structure while other presented themselves very heterogeneously. For instance, G1 has a very similar structure for its pages, but depending on the section where the story is put into, the site design changes slightly, most in colors and images, controlled by style sheets. Another example of this was Estadão, which presented a similar behaviour, but with structural changes for some of its sections, despite the visual similarity. While most news portals give each section its own subdomain, which makes it easier to differentiate their designs, Terra has several subdomains but keeps their designs exactly the same.

On the other hand, some websites presented themselves as a challenge to annotate automatically. We attempted to use the USATODAY news portal, but its sections were all structurally different from one another and their URLs very similar, which makes it very hard to distinguish how to annotate automatically. Because of this, we chose to discard it.

It is interesting to note that these problems are not limited to the step of obtaining a corpus, but are also present when reporting results, since the distribution of pages from that website can play a big part in the overall results. For this reason, if considerable differences are found between pages of a same design, we gave preference to those that could be found more easily, in an attempt to measure how well we would perform in the majority of cases for that site.

5.1.2 Annotation format and process

There are several ways to create annotations for webpages. Possible ways include storing the XPath [31] to the relevant nodes, recreating the page's structure in an auxiliary file with annotations and adding an annotation attribute to HTML element nodes, among others.

The first two methods are less intrusive, since they do not modify the original document. The last one, however, is easier to manipulate since once

the document is parsed, all annotation information is readily available and annotations are guaranteed to be up-to-date with the latest version of the document. After some experimentation, we concluded that modifications to the original document eventually introduced during annotation have a negligible impact. Thus, this is the method we have chosen for our corpora.

It is relevant to note some limitations to this format: since we are interested in the content on text nodes and we can only annotate element nodes, eventually our annotations might not perfectly map to the desired content. For instance, consider the example HTML snippet and the associated DOM tree described in Figure 5.1. To annotate the publication date ("today, 5 hours ago"), we would have to annotate the `div` element. This, however, would include several other text nodes that are not of interest. One way to mitigate the impacts of this is the use of a "`other`" label, which can be applied to the `b` nodes, but which can't avoid the annotation of "John" as part of the publication date.

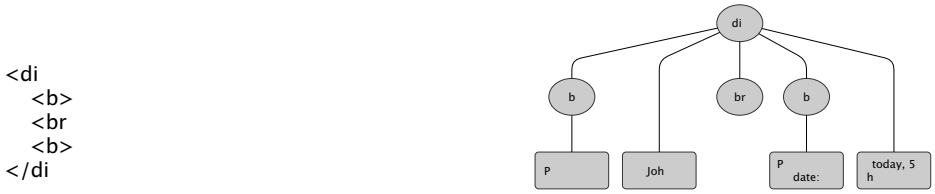

Figure 5.1: Example of a potential weakness of annotating element nodes.

The annotation format originally used by NEWS600 uses an auxiliary XML file, recreating the DOM tree structure, which can annotate individual text nodes. However, there are still some problematic cases that this format is not immune to, such as the one depicted in Figure 5.2, where both author and publication date information are placed in the same text node. The solution for cases like these, in both formats, is to use a compound label such as "`author_and_date`".

Figure 5.2: Example of a shared weakness of element node and text node annotations.

Once settled with the format, we created a visual tool to aid in manual annotations of webpages. For that, we used Firebug [9], an extension for the Mozilla Firefox web browser [10], as our starting point and modified it to our needs. We leveraged the "Inspect" functionality of Firebug to select a DOM node to add our annotation attribute, saving the webpage to disk afterwards. A screenshot of our tool can be seen in Figure 5.3.

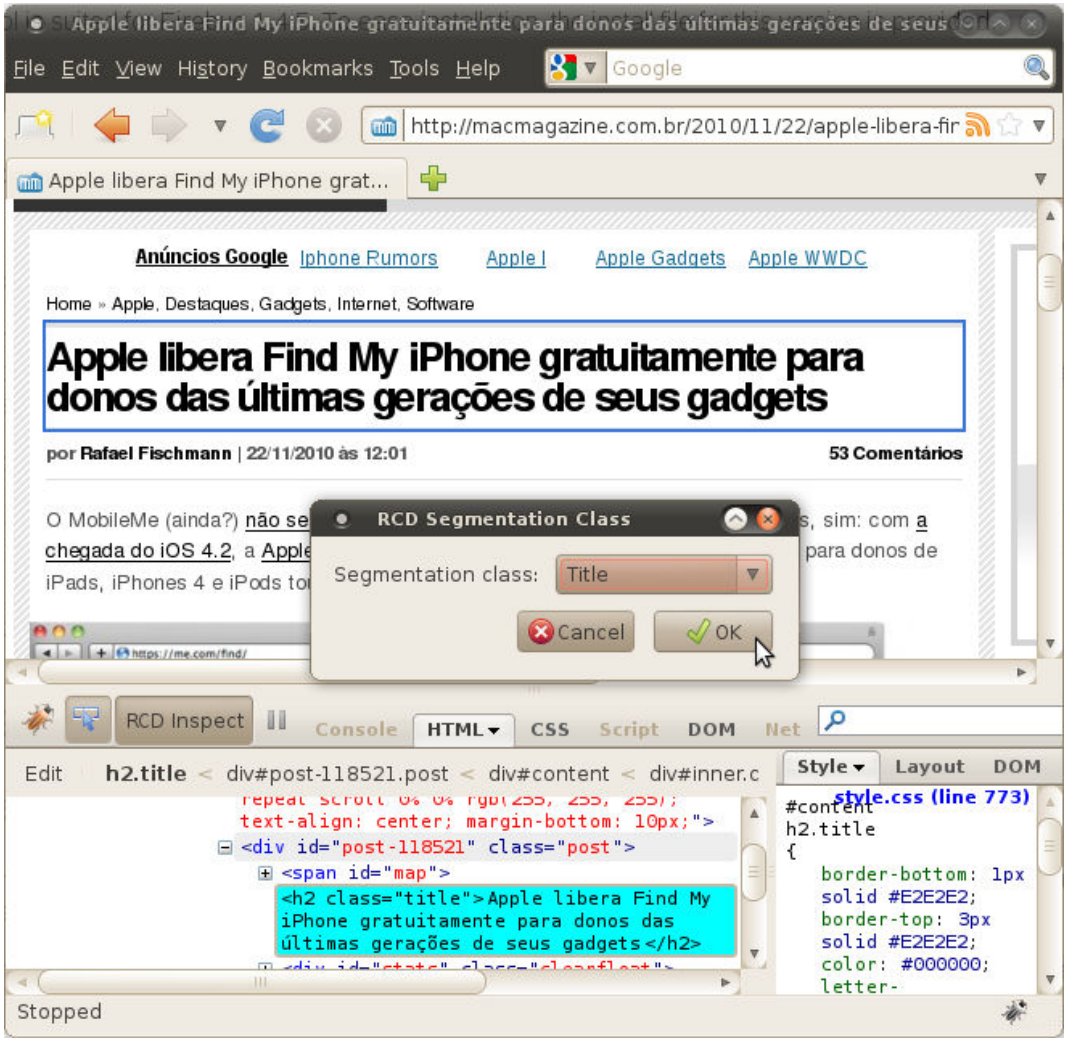

Figure 5.3: Screenshot of our tool for visual annotation of webpages.

5.2 Evaluating works in the literature

In this section we evaluate the works described in Chapter 4. We will also use them as a basis of comparison for the results of our approach.

5.2.1
Luo, et al. (2009)

The work of Luo, et al. (2009) [22] and the details of our implementation of it are explained in Section 4.1. Since this work does not require training, reported results are averages of the whole corpora. Unfortunately, they do not provide results for each news site, but report an overall result of 91.571% of precision and 99.145% of recall. The metric used by them is segment-based, which they claim to be stricter than bags of words, and thus more susceptible to sudden fluctuations in the results. We, however, still opted to use bag of words, which is described in Section 2.5.2.

Results of our own implementation is available in Table 5.1, divided into *full rendering*, which uses the WebKit rendering engine, and *partial rendering*, which uses an HTML parser combined with our simplified CSS parser. We did a breakdown of the Cardoso3000 corpora by news site, along with an overall value. Precision and recall are calculated as averages from each document and F1 is calculated using the reported precision and recall. Sites of particular interest are CNN, BBC and NYTimes, as they also appear in the original work's corpus.

	Full rendering			Partial rendering		
News portal	Prec.	Recall	F1	Prec.	Recall	F1
BBC	0.94	0.43	0.59	0.99	0.44	0.61
CNN	0.96	0.79	0.87	0.96	0.86	0.91
Estadão	0.94	0.59	0.73	0.93	0.61	0.73
Folha	0.98	0.45	0.61	0.97	0.44	0.61
FOXNEWS	0.95	0.36	0.52	0.99	0.44	0.62
G1	0.68	0.50	0.58	0.56	0.99	0.71
NEWSWEEK	1.00	0.97	0.99	1.00	0.97	0.99
NYTimes	0.95	0.94	0.94	0.78	0.99	0.87
Terra	0.87	0.87	0.87	0.90	0.82	0.86
TIME	0.88	0.84	0.86	0.88	0.83	0.85
Cardoso3000 corpus	**0.92**	**0.67**	**0.78**	**0.90**	**0.74**	**0.81**

Table 5.1: Results of our implementation of Luo, et al. (2009) for full and partial rendering approaches.

Our results are not on par with those reported, with the exception of NEWSWEEK and NYTimes. Since the biggest difference is in recall, rather than precision, we rule out the possibility that it is caused by the lack of the postprocessing step. It could be due to some preprocessing of the webpage, non-published details for their approach or some misinterpreted subtlety of their method.

We observe a slight improvement with the partial rendering for some sites, but it would be surprising for our simplified CSS parser to be better than a standards-compliant rendering engine. We find it easier to attribute the small fluctuation in results to the HTML parser used, which differs for each implementation. It is not exactly a matter of one being better than another, since both WebKit and libxml2 are established as being of very high quality, but due to subtleties that could generate some structural differences in DOM trees.

5.2.2 Wang, et al. (2009)

The work of Wang, et al. (2009) [40, 41] and the details of our implementation of it are explained in Section 4.2. Differently from how the authors approached the problem, we did not create one model per domain to apply to all others. Instead, our model is trained with 100 pages from each news site in the Cardoso3000 corpora, and we applied it to the rest of the pages. This is a slightly more optimistic scenario, as the model should be better than one trained in a single site. Like the original work, we used an SVM model with RBF kernel, but we haven't invested in tuning any parameters, keeping them at the default values provided by Weka [12].

As previously mentioned when we described their work, the idea is to identify a node containing the news body and a node containing the news title. However, as the node containing the body is not always exempt from uninteresting content, such as links to related stories, user comments, etc., we decided to evaluate how well it would perform when considering such scenarios.

We will measure the results of our experiments by two metrics: *node-based* and *content-based*. Both use bag of words, as described in Section 2.5.2, but differ in how the body and title nodes are annotated for evaluation. The former attempts to mimic their annotation, which identifies a single node and considers its whole subtree as body, and is always used for training the machine-learning models; the latter is the one we will be using to evaluate our approach, which differentiates the news body from other content possibly present in the identified subtree. Results of our implementation of their work are available in Table 5.2.

It is interesting to note what happens on the body detection for portals such as TIME, shared by theirs and our corpus, when using different metrics. Unlike the majority of these news sites, in which a single subtree can accurately represent the news body, portals such as Folha, NYTimes and TIME include some amount of uninteresting content, which explains the difference in the

	Node-based			Content-based		
News portal	Prec.	Recall	F1	Prec.	Recall	F1
BBC	0.98	1.00	0.99	0.98	1.00	0.99
CNN	0.97	0.98	0.97	0.97	0.98	0.97
Estadão	0.99	0.99	0.99	0.99	0.99	0.99
Folha	0.79	0.99	0.88	0.49	0.99	0.66
FOXNEWS	0.99	0.99	0.99	0.99	0.99	0.99
G1	0.97	0.95	0.96	0.97	0.95	0.96
NEWSWEEK	0.95	0.48	0.63	0.95	0.48	0.63
NYTimes	0.63	0.98	0.77	0.57	0.99	0.73
Terra	0.79	0.91	0.85	0.79	0.91	0.85
TIME	1.00	0.99	0.99	0.90	0.99	0.94
Cardoso3000 corpus	**0.91**	**0.93**	**0.92**	**0.86**	**0.93**	**0.89**

Table 5.2: Results for body detection.

results for both metrics. Similarly, because the method correctly identifies the body nodes for the majority of pages, for the other domains it achieves 100% on these pages regardless of metric, explaining the identical results observed with different metrics.

However, the impact we see on body for some sites does not appear to happen with titles, probably due to the simplicity of the title DOM subtrees, as we can see in Table 5.3.

Overall, in most sites shared between theirs and our corpus, such as BBC, CNN, FOXNEWS and TIME, our results are close to their reported value of 98.1%, with NEWSWEEK being the only exception. We are not sure why this happens, but given the good results with other news sites, it is possibly due to some details in annotation or page design, which may have changed in the interim from publication to our implementation. Looking at results, we believe that we manage to reproduce the results of this method reasonably well.

5.2.3 Established basis of comparison

Because the results for the first approach were not satisfactory, we will only be considering the results of Wang, et al. (2009) [40, 41] as a basis of comparison for our work. In Table 5.4 we review these numbers, adopting the content-based bag of words metric.

	Node-based			Content-based		
News portal	Prec.	Recall	F1	Prec.	Recall	F1
BBC	1.00	1.00	1.00	1.00	1.00	1.00
CNN	1.00	1.00	1.00	1.00	1.00	1.00
Estadão	1.00	1.00	1.00	1.00	1.00	1.00
Folha	0.97	0.97	0.97	0.97	0.97	0.97
FOXNEWS	1.00	1.00	1.00	1.00	1.00	1.00
G1	0.99	0.99	0.99	1.00	1.00	1.00
NEWSWEEK	1.00	1.00	1.00	1.00	1.00	1.00
NYTimes	0.05	0.00	0.01	0.05	0.00	0.01
Terra	0.99	0.84	0.91	0.99	0.84	0.91
TIME	1.00	1.00	1.00	1.00	1.00	1.00
Cardoso3000 corpus	**0.90**	**0.88**	**0.89**	**0.90**	**0.88**	**0.89**

Table 5.3: Results for title detection.

Task	Precision	Recall	F1
Title detection	0.90	0.88	0.89
Body detection	0.86	0.93	0.89

Table 5.4: Basis of comparison for our approach; results from our implementation of Wang et al. (2009) [40, 41] on the Cardoso3000 corpus, measured with a content-based bag of words metric.

5.3 Our approach

In this section, we will denote by *extra-site cross-validation* an experiment that trains a model on a set of domains different from those of the documents used for testing; that is, folds are domain-disjoint. Similarly, we denote by *intra-site cross-validation* an experiment that trains and tests on pages from the same domain; that is, folds are representative of the domain distribution in the corpus.

We start by extracting the relevant content of the webpage with the NCE algorithm described in [18], using it as a base for the next steps. This reduces the amount of nodes that we need to classify in order to find the news title and publication date.

Our experiments show that a 35 to 40% speedup in total execution time is obtained by discarding the nodes that are not considered part of the relevant content. These discarded nodes have affected the quality of our results in at most a 1% decrease in F1. For a method that focuses on large-scale document processing, this performance boost is a must, and will be described in more details in the next section.

Unless otherwise noted, the corpus used for the experiments is the RCD4 corpus, described in Section 5.1.1.

5.3.1 Title detection

We conducted a series of experiments to evaluate the impact of visual presentation attributes. As a baseline for comparison, we use a classifier that always determines the document's `<title>` tag as the document title. The results for strictly structural features and our combined approach can be seen in Tables 5.5 and 5.6.

Method	Precision	Recall	F1
Baseline (`<title>` tag)	0.61	0.89	0.72
3-fold cross-validation (extra-site)	0.72	0.90	0.80
3-fold cross-validation (intra-site)	0.73	0.91	0.81

Table 5.5: Title extraction results on the RCD4 corpus for strictly structural attributes

Method	Precision	Recall	F1
Baseline (`<title>` tag)	0.61	0.89	0.72
3-fold cross-validation (extra-site)	0.88	0.95	0.91
3-fold cross-validation (intra-site)	0.90	0.94	0.92

Table 5.6: Title extraction results on the RCD4 corpus for both structural and visual attributes

We observe that the use of visual presentation attributes significantly improves our results, with over 10% increase in F1.

A time comparison of these approaches is given in Table 5.7, measured relatively to the structural approach. As an example, the value 1.77 for the Structural + Visual method indicates that it is 77% slower than the strictly structural approach. The table also includes timings for the approach that skips the relevant content detection step, thus classifying every node in the DOM tree, which we identify as "whole tree", and time estimates for an equivalent approach that uses full rendering instead of our simplified CSS parser. The rendering engine used was WebKit [42], with scripts and plugins disabled. We then proceeded to add to the rendering time of the pages the average time it took for our algorithm to run once every feature is computed. We toggled the use of images, as they might be of interest to preserve the appearance of the webpage in case geometric positioning of nodes is needed.

Method	Time taken
Baseline (`<title>` tag)	0.40
Strictly structural	1.00
Structural + Visual	1.77
Structural + Visual, whole tree	3.01
WebKit rendering, no images	10.80
WebKit rendering	39.16

Table 5.7: Title extraction execution times on the RCD4 corpus, relative to strictly structural approach

5.3.2 Date detection

Our model for date detection is title-dependent. That is, it depends on a correct classification of the title because this information is used for attribute 8 (see Section 3.2) during date detection. However, when a title node is not classified, this dependency will most likely prevent our models from obtaining a correct classification.

We then employed two different models, depending on the title detection outcome. If some node is classified as title in the previous step, we proceed with the title-dependent model for dates. Otherwise, a title-independent model is used. Results for the conditional and title-dependent approaches are shown, respectively, in Tables 5.8 and 5.9.

Method	Precision	Recall	F1
3-fold cross-validation (extra-site)	0.88	0.66	0.75
3-fold cross-validation (intra-site)	0.88	0.82	0.85

Table 5.8: Date extraction results on the RCD4 corpus for the conditional approach with two models, after post-processing

Method	Precision	Recall	F1
3-fold cross-validation (extra-site)	0.79	0.75	0.77
3-fold cross-validation (intra-site)	0.85	0.83	0.84

Table 5.9: Date extraction results on the RCD4 corpus for the title-dependent model, after post-processing

Surprisingly, we observe that the title-dependent model still seems to be the best choice as it produces more balanced results for precision and recall, specially for the extra-site cross-validation.

5.3.3
Body detection

For body detection, we consider every node returned from the NCE algorithm of [18] used at the start of the pipeline, excluding only those that were detected as title or publication date. As only a few nodes were excluded from the returned set, the results were unaffected, as shown in Table 5.10.

Method	Precision	Recall	F1
NCE of [18]	0.82	0.92	0.87
After our pipeline	0.82	0.92	0.87

Table 5.10: Body extraction results on the RCD4 corpus

We experimented with a post-processing stage that would also consider all nodes between the title and the first detected body node, but the results weren't very revealing: very little have changed, and often times for the worse.

5.3.4
Validation of results

We refer back to the results we outlined in Section 5.2.3, and will compare them to those obtained with our approach.

We will complement the results already reported with the execution of our approach using the Cardoso3000 corpus (Section 5.1.1). When applying our method on Cardoso3000, the model is trained with the entire RCD4 corpus. We also applied on the whole of RCD4 with a model trained with Cardoso3000. Because of the shared domains between both corpora, this is considered an intra-site execution.

We consolidate the results we obtained for title detection in Table 5.11 and body detection in Table 5.12. Results are ordered by increasing F-measure, then increasing precision. We denote by "Wang" our implemented approach from Section 5.2.2.

Approach	Training	Test	Precision	Recall	F1
Wang	Cardoso3000	RCD4	0.84	0.75	0.79
Ours	RCD4	Cardoso3000	0.84	0.93	0.88
Ours	Cardoso3000	RCD4	0.84	0.95	0.89
Wang (hold-out)	Cardoso3000	Cardoso3000	0.90	0.88	0.89
Ours (intra-site CV)	RCD4	RCD4	0.90	0.94	0.92
Ours (hold-out)	Cardoso3000	Cardoso3000	0.91	0.98	0.94

Table 5.11: Results validation for the task of title detection.

Approach	Training	Test	Precision	Recall	F1
Wang	Cardoso3000	RCD4	0.73	0.92	0.81
Ours (intra-site CV)	RCD4	RCD4	0.82	0.92	0.87
Ours	Cardoso3000	RCD4	0.83	0.92	0.87
Ours	RCD4	Cardoso3000	0.87	0.90	0.88
Wang (hold-out)	Cardoso3000	Cardoso3000	0.86	0.93	0.89
Ours (hold-out)	Cardoso3000	Cardoso3000	0.87	0.90	0.89

Table 5.12: Results validation for the task of body detection.

In Table 5.13 we show results for date detection using the Cardoso3000 corpus for training and testing, and in Table 5.14 we show detailed results for date detection by site for the Cardoso3000 corpus, with a model trained from RCD4 for a better understanding of the results observed.

Approach	Training	Test	Precision	Recall	F1
Ours	Cardoso3000	RCD4	0.95	0.55	0.70
Ours	RCD4	Cardoso3000	0.94	0.59	0.72
Ours (hold-out)	Cardoso3000	Cardoso3000	0.99	0.62	0.76
Ours (intra-site CV)	RCD4	RCD4	0.87	0.85	0.86

Table 5.13: Results validation for the task of date detection.

We have also applied our methods to the NEWS600 corpus [34], described in Section 5.1.1. We trained our model on the provided training and validation sets, composed of 304 pages, and applied to the provided test set, with 300 pages total. The results obtained are shown in Table 5.15, along with the best published results we have found for the same corpus, gathered from the works of [33, 34, 35]. These works make use of full rendering to extract a wide range of hundreds or thousands of features, based on content, structure, page geometry and task-specific features, such as checking whether a text node matches a regular expression for date detection. In particular, the work of [35] creates a graph combining the structural ordering of elements and their rendered positions, which is used in addition to the previously mentioned features. Our approach is measured using bag of words, while published results use an exact node metric, as described in Section 2.5.

Looking at the results, we believe our approach to be on par with the compared works. The use of different corpora support the stability of our results, which are not limited to the corpus used during development. Also, it is important to stress the simplicity of our features and that we do not perform a full rendering of the page, which grants us much more competitive execution times.

News portal	Prec.	Recall	F1
BBC	0.93	0.00	0.01
CNN	0.99	0.89	0.94
Estadão	0.99	0.92	0.96
Folha	0.98	0.61	0.75
FOXNEWS	0.99	0.96	0.98
G1	0.66	0.95	0.78
NEWSWEEK	0.97	0.01	0.01
NYTimes	0.99	0.27	0.42
Terra	0.95	0.99	0.97
TIME	0.93	0.26	0.40
Cardoso3000 corpus	**0.94**	**0.59**	**0.72**

Table 5.14: Results validation for the task of date detection.

Approach	Label	Precision	Recall	F1
Baseline (`<title>`)	Title	0.63	0.93	0.75
Our approach	Title	0.90	0.96	0.93
SVM of [33]	Title	0.99	0.96	0.97
Our approach	Date	0.87	0.85	0.86
CRF of [35]	Date	0.83	0.92	0.87
Our approach	Body	0.82	0.95	0.88
SVM of [33]	Body	0.90	0.98	0.94

Table 5.15: Results on the NEWS600 corpus, ordered by F1. The baseline and our results are measured using bag of words.