4
Evaluation of selected works in the literature

We have selected two interesting works in the literature to implement and evaluate: Luo, et al. (2009) [22] and Wang, et al. (2009) [40, 41]. Both are recent and easy-to-understand works that present very good results. The first detects the news body using a technique similar to node clustering; the second makes use of the page's geometry to aid in the detection of news title and body.

In the following sections we will describe them and in Chapter 5 we will compare reported results with those obtained.

4.1
Luo, et al. (2009)

The work of Luo, et al. (2009) [22] attempts to detect the body text of a news page. It uses the Maximum Scoring Subsequence (MSS) approach from [28] with a simple scoring function based on presentational attributes of the text.

The Maximum Scoring Subsequence (MSS) problem is defined as: given a sequence $(x_1, x_2, \ldots, x_n)$ of real numbers, find the contiguous subsequence $(x_i, \ldots, x_j)$ with $i \leq j$ such that the sum of its values is maximal in the sequence. This can be calculated in linear time using a dynamic programming algorithm [32].

The page's text is broken into text segments, and a scoring function is applied to them in order to create a sequence for the MSS algorithm. The maximum scoring subsequence is then mapped back to text segments in order to identify the body text of the news page.

A text segment is analogous to a paragraph: a unit of text vertically separated from the rest. A common problem with DOM text nodes is that they may be segmented due to the use of formatting elements, such as those introducing emphasis, or links. We illustrate how this might happen in Figure 4.1. Text segments groups these types of text nodes creating a single unit of text. This is done by looking at the `display` CSS property: a value of `block` indicates a line-break before and after the element, so every node

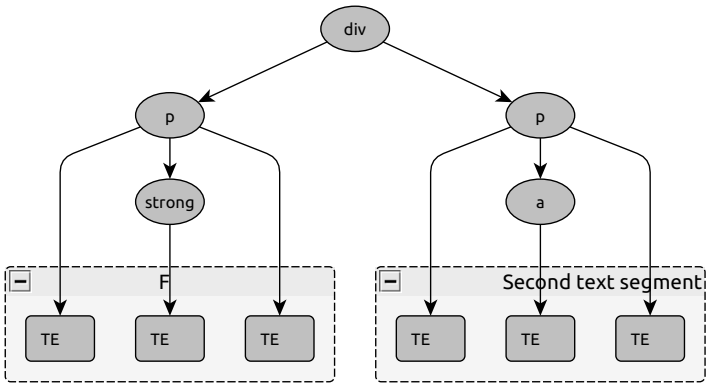

Figure 4.1: Example of using DOM text nodes versus text segments.

between two block elements are grouped into a text segment, even if a single-node segment is to be formed this way.

Let s be a text segment and p_{size}, p_{color} and p_{link} be the percentage of the segment's text that shares the document's most common font size, most common color and is contained in links, respectively. The score of s is given by $F(s) \times StringLength(s)$, with $F(s)$ defined as:

$$F(s) = \begin{cases} 1 & p_{size} \geq 0.7,\ p_{color} \geq 0.2,\ p_{link} \leq 0.5 \\ -1 & \text{otherwise} \end{cases}$$

In the original work, they do some postprocessing to the selected text segments in order to remove images and sidebars with related links or similar content which are not directly part of the news story. To do this, they look at the bounding box of text segments. The segments of the article body are aligned and tend to share the same left positioning and width, which are used to remove irrelevant content. However, the authors do not go into details of how this postprocessing works. For instance, an image might be sharing screen space with the news in some portions of the story, like depicted in Figure 4.2, which makes it less obvious to filter and remove. We have then chosen not to implement this step and, because of that, our results might be lower than expected.

The authors suggest that a local classifier can be used for $F(s)$, but their provided values, chosen heuristically, already gives very good results: they report 91.6% of precision and 99.1% of recall of text segments. They also claim that measuring precision and recall in terms of text segments is more strict than measuring in terms of words, in a similar fashion of how a node-based metric might be more strict than bag of words.

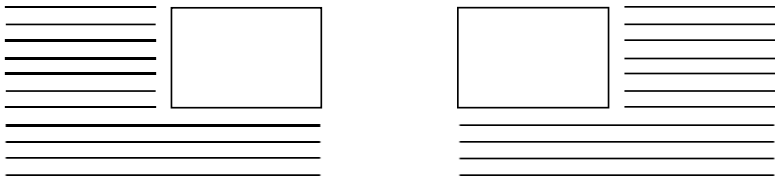

Figure 4.2: Examples of non-obvious cases for the postprocessing step of [22].

They also acknowledge the need for rendering the webpage to compute the CSS property that aids in segment identification. However, since we do not make use of the postprocessing stage, which requires geometric information from the rendered webpage, we believe our simplified CSS parser can be used for this task as well.

4.2 Wang, et al. (2009)

The work of Wang, et al. (2009) [40, 41] attempts to detect the title and story body of a news page. It requires a full rendering of the page to obtain geometric positioning of nodes and uses a machine-learning model to classify them as title and body.

To detect the news body, it searches for nodes with a great portion of the page's formatting elements containing a great portion of the page's formatted content and a large screen area. For titles, the nodes ideally have a long narrow bounding box with short vertical distance and high horizontal overlap to the body, no full stop at the end of sentence and a small number of words. To identify these node they use a nonlinear SVM model with a Gaussian RBF kernel. No details were given with respect to the tuning of these models.

The author's premise is that the pages can be very long, but the title and body are always in the beginning of the document. The notion of distance from the beginning of the document is measured in *screens*: content shown without scrolling the page is considered to be the *first screen*; scrolling the page past the first screen will reveal the *second screen* and so forth. Title features are normalized in respect to elements rendered in the first screen, while body features are normalized in respect to elements in the first two screens.

While the notion of screens is an interesting approach to the problem, not all screens are the same: some are bigger, allowing higher resolutions to be used and more content to be displayed at once; some are smaller, requiring the use of lower resolutions and thus less content; some users are visually impaired and increase the font size of webpages, reducing the total content shown in each screen; etc. Since the authors do not mention the screen size considered,

we assume a screen size of 1366x768, based on the popularity of this resolution in modern notebooks[1], and websites displayed in full screen, without loss of area by features of the web browser and operating system. Ideally, websites are designed with usability and different screen sizes and resolutions in mind, so hopefully this should not be a big issue and no different screen sizes were explored.

The notion of horizontal overlapping and vertical distance is not described in details by the authors. We know what they are meant to represent and how they are normalized, but the vertical distance and horizontal overlap area is not well defined, and could be open to interpretations. For instance, one could interpret the vertical distance as an absolute value, while other might consider it as a signed value; that is, nodes below the body having a negative distance. To be clear, our implementation of their work assumes the following:

- the vertical distance is defined as the (signed) distance, in pixels, from the bottom of the title node to the top of the body node, normalized by the screen height; and
- the absolute horizontal overlap is defined as the horizontal area of the screen, in pixels, shared by the title and body nodes, normalized by the screen width.

To illustrate, we show in Figure 4.3 the dimensions considered for each feature.

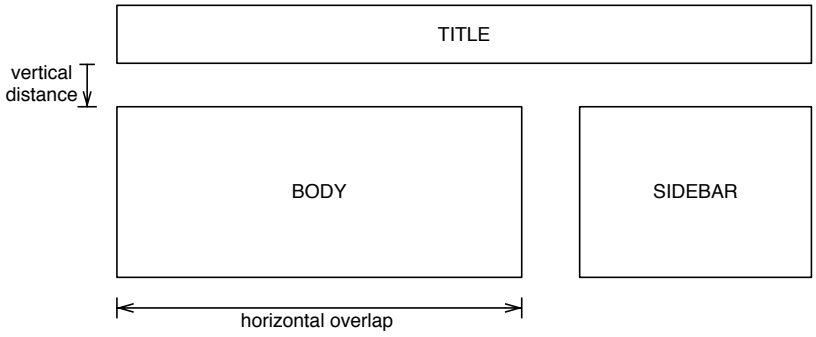

Figure 4.3: Illustration of the dimensions considered for the vertical distance and horizontal overlap features for title nodes.

Another interesting contribution of this work is their approach of training a model on a single site, which is then applied to all others. This was used to illustrate the generality of their method and how well it adapts to unseen domains; an issue seldomly taken into account in the literature. Also, they experimented with different sizes of training sets in an attempt to understand

[1]This resolution is usually found in 15" monitors with a 16:9 aspect ratio.

how many examples are needed to learn a good model. Their best model, trained using only pages from the USNEWS website, achieved 98.1% *accuracy* for their entire corpus, which includes sites such as BBC, CNN, FOXNEWS, NEWSWEEK and TIME, among others. They calculate accuracy as the number of documents with a correctly identified body and title pair over the total number of documents.

One criticism we have to their approach is that they attempt to detect a single node containing the body and perform no postprocessing afterwards. This means that some uninteresting parts of the page, such as related stories, printing and sharing links, are obtained and considered part of the body, which is not reasonable. We present an example of this issue in Figure 4.4, where it is easy to see that, when using a content-based metric such as bag of words, their result is subject to some variation.

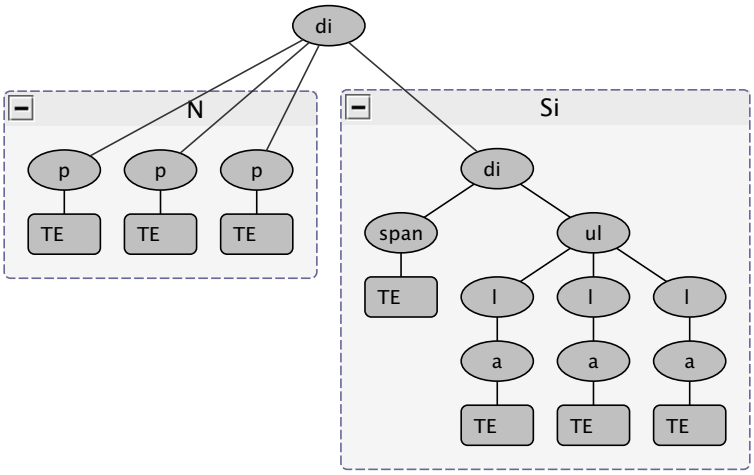

Figure 4.4: Illustration of potential issues with their body node classification; notice how the node that contains the news body also contains unimportant content.