

Learning to Extract Content from News Webpages

Alex Spengler and Patrick Gallinari
Laboratoire d'Informatique
Université Pierre et Marie Curie
Paris, France

Abstract

We consider the problem of content extraction from on-line news webpages. To explore to what extent the syntactic markup and the visual structure of a webpage facilitate the extraction of its content, we compare two state-of-the-art classifiers as first instantiations of a general framework that allows for proper model comparison. To this end, we introduce the publicly available NEWS600 corpus, a set of 604 real world news webpages which have been annotated with 30 semantic labels. An empirical analysis of the two models on this dataset shows that the inclusion of structural information is indeed advantageous.

1. Introduction

With the rising popularity of the web, the creation and management of content has considerably evolved and its presentation is getting more and more elaborate. Nowadays, webpages exhibit a large variety of visually and semantically distinguishable regions. There might be navigation menus, banners and various types of advertisements scattered all over the page, polls, links to related pages, copyright notices and contact information, comment boxes and, of course, the actual content of the page itself. Furthermore, the main content might again be split into multiple regions. A web news article, for instance, usually contains the story title, its author, the date of publication, multimedia and links to related articles. Although each of these pieces of information serves a specific semantic purpose, most of them essentially guarantee the functioning of the site and are of no or minor relevance to the primary content of the webpage.

The accurate extraction of primary content, however, facilitates a number of tasks, ranging from automatic text summarization over speech rendering for the visually impaired to information retrieval. The identification of content regions is equally crucial to the restructuring of webpages for mobile devices with constrained screen space, such as smart phones and PDAs. Several papers [7, 4, 3, 12, 11]

have recognized the importance of content extraction and proposed solutions to this problem. Most of these approaches are based on a fixed set of complex, handcrafted heuristics and typically involve several, successive processing steps. While this might work well on a restricted set of pages for very specific tasks, it has some shortcomings. First, due to the intrinsic variability in syntactical markup and visual structure, it is difficult and costly to adapt the individual heuristics to previously unseen pages. Second, greedy selection and fixed combination of heuristics are both likely to produce suboptimal solutions. Third, pipelined processing steps result in cascading errors, so that a recovery from early mistakes becomes non-trivial. In addition, the comparison of different approaches is clearly hampered by the unavailability of a publicly available dataset.

CONTRIBUTIONS. In this paper we present a principled approach to the problem of automatic content extraction from a set of heterogeneous, but domain-oriented webpages. Our contributions are:

1. We formulate the problem in a classification framework that can be applied to any webpage and any extraction task and allows for accurate model comparison. Surprisingly, this has not been done before. Then, in order to explore to what extent the inherent syntactic and visual structure of a webpage facilitates the extraction of its content, we employ two state-of-the-art classifiers as first instantiations of the framework: A multi-class support vector machine (SVM) and a sequential conditional random field (CRF). Both automatically learn to weigh the importance of thousands of features, integrating content, style and structural knowledge into a single probabilistic model (section 2).
2. We introduce the NEWS600 corpus. It comprises 604 heterogeneous news webpages that have been annotated with 30 distinct semantic labels. It can be used for several tasks, including title extraction and advertisement removal and is publicly available from <http://webia.lip6.fr/~spengler/news600/> (section 3).

3. We apply the SVM and the CRF to the NEWS600 dataset and evaluate their overall and individual label performances. The results show that the CRF, which integrates the implicit sequential structure of a webpage, outperforms the SVM. However, it is not obvious to decide what information or features to include a priori (section 4).

2. Web content extraction

The Document Object Model (DOM) defines a common interface to a webpage. It provides logical access to the content, structure and style of a page through an ordered tree of element, text, comment and attribute nodes. Although each node in the DOM tree contains vital information, it is only in the element and text nodes where the actual content of a webpage can be found. Attributes and comments are not visible to the end-user when the page is rendered in a browser. Moreover, the visible content of a webpage lies in essence in the leaves of the DOM tree. The extraction of content can thus be formulated as a classification problem in which we seek to assign the right semantic label to each visible leaf node in the DOM tree.

2.1. Framework

In general, we hence propose to approach the content extraction problem in a classification framework. To this end, we construct an objective function on a subset of the DOM nodes—typically, but not necessarily, the visible leaf nodes—such that the optimization of this function results in a good semantic labeling. Although we do not impose any constraints on the concrete form of the objective function, it might encode the cost of a particular label configuration. Furthermore, it should combine all relevant sources of information that contribute to an accurate labeling while remaining computationally feasible. Please note also that the proposed framework does not oblige us to work in a supervised setting in which we are given the correct labeling.

To express the problem of content extraction in the described classification framework offers several advantages. First, it is general and can be applied to any webpage and any extraction task. Second, it clearly separates the algorithm from the data, so that it is easier to disseminate corpora and compare different approaches. Third, because there is no need to compare extracted pieces of HTML using text similarity measures such as the edit distance, it allows for a more accurate performance evaluation that renders algorithmic comparison more reliable. Fourth, classification is a well-studied problem in machine learning, providing both a theoretical foundation and best practices.

2.2. Support vector machines and conditional random fields

As first instantiations of the afore-mentioned classification framework for content extraction, we employ two specific discriminative models, the SVM and the CRF. Both belong to the class of *generalised linear models* (GLIM), a family of classifiers h which is linear in the parameters w :

$$h_w(\mathbf{x}) = \arg \max_{\mathbf{y} \in \mathcal{Y}} \langle \mathbf{w}, \phi(\mathbf{x}, \mathbf{y}) \rangle,$$

where $\mathbf{y} = (y_1, \dots, y_K)^T$ is a particular labeling (of the visible leaf nodes) of a webpage \mathbf{x} and ϕ is a vector-valued function of features (cf. section 2.3). Acting on both \mathbf{x} and \mathbf{y} , ϕ enables us to encode a wide variety of interdependencies in the input-output space and thus offers an easy way to balance modeling freedom and mathematical tractability.

In order to measure the error of a classifier h on a new labeled webpage (\mathbf{x}, \mathbf{y}) , we avail ourselves of a *loss function* $\ell(\mathbf{x}, \mathbf{y}, h(\mathbf{x}))$. It quantifies the discrepancy of the prediction $h(\mathbf{x})$ and the correct labeling \mathbf{y} , given the corresponding input \mathbf{x} . Statistical learning theory shows that the classifier h that minimizes the *regularized empirical risk*

$$\mathcal{R}_{\text{reg}}^\ell[h] = \frac{\lambda}{2} \|\mathbf{w}\|^2 + \frac{1}{N} \sum_{i=1}^N \ell(\mathbf{x}_i, \mathbf{y}_i, h(\mathbf{x}_i)) \quad (1)$$

also minimizes the true mean error on a previously unseen document \mathbf{x} . One of the crucial differences of the SVM and the CRF concerns the concrete form of the loss function ℓ . The first uses the non-differentiable hinge-loss, so that (1) becomes a quadratic program. The latter uses the differentiable log-loss, meaning that (1) can generally be optimized by the maximum likelihood principle. More important to our discussion here, however, is the way the labeling score $\langle \mathbf{w}, \phi(\mathbf{x}, \mathbf{y}) \rangle$ is factorized. The multi-class SVM we employ predicts each semantic label y_k independently of its neighbours, i.e. $\langle \mathbf{w}, \phi(\mathbf{x}, \mathbf{y}) \rangle = \sum_{k=1}^K \langle \mathbf{w}_k, \phi(\mathbf{x}, y_k) \rangle$. The sequential CRF [6] additionally incorporates transition features between neighbouring nodes y_k and y_{k+1} , thus making a first-order Markov assumption: $\langle \mathbf{w}, \phi(\mathbf{x}, \mathbf{y}) \rangle = \sum_{k=1}^K \langle \mathbf{w}_k, \phi(\mathbf{x}, y_k) \rangle + \sum_{j=1}^{K-1} \langle \mathbf{w}_{j,j+1}, \phi(\mathbf{x}, y_j, y_{j+1}) \rangle$. It is the inclusion of label interdependencies via transition features that constitutes the major difference between the employed CRF and SVM, and not the slightly different form of the loss functions.

2.3. Features

We utilize a large variety of characteristics which help to predict a semantic label for a visible leaf node. For each word in the node's text content, for each attribute value, each DOM parent node name, each surrounding comment

etc. we create a feature. We also create features from DOM information, such as the number of siblings, the DOM tree depth and whether the node is contained within an `iframe` or not. In addition to this, we include over 300 style and layout features all of which we recover from the browser's rendering engine. Examples are the position, width and height of a node, its background color, padding, margin and border as well as its font size and color (if applicable). Moreover, we compute some features based on document-wide measurements, such as the difference between the node's font size and the maximal font size of the page. Finally, we employ a small number of regular expressions that capture dates, weekdays, time strings and so on. In total, there are 108,692 features (per label). Please note also that the transition features of the CRF depend on the labels of two neighbouring nodes only and that our current model does not use conjunctions of the afore-mentioned atomic features.

3. The NEWS600 webpage corpus

On the following lines, we present the NEWS600 webpage corpus. It contains 604 real world news webpages that have been accurately annotated with 30 distinct semantic labels. The corpus can be downloaded from <http://webia.lip6.fr/~spengler/news600/>.

3.1. Collection

As indicated, the goal was to collect a set of webpages that is domain-oriented, yet heterogeneous. To this end, we chose online news articles as application domain because it allows (limited) comparison with prior work. We thus selected more than 600 news article URLs from a large number of different web domains. As we used RSS news feeds to collect the URLs, the number of pages per domain roughly reflects the popularity of the corresponding news service. All pages we collected are written in English, yet contain a huge variety of news categories from numerous anglophone countries. After the removal of duplicates and pages with inaccessible content, the final set consisted of 604 news article URLs, all of which are part of the NEWS600 corpus. The URLs have been gathered in March 2008 over a range of three days.

Having selected the URLs of interest, their contents needed to be saved. At this point it is crucial to ensure that no valuable information is lost, so that a precise offline reconstruction of the page will be available. Amongst other details, remote documents, which are automatically loaded into `iframe` elements, need to be stored with the least possible number of modifications to the original page. We also saved images and animations. This is important to achieve a realistic rendering of an offline page. Clearly, the storage of all relevant CSS files is hence obligatory.

3.2. Annotation

The annotation process has to be treated with equal diligence. One observation of importance in this regard is the fact that most of the content of a webpage is located in the leaf nodes of its DOM tree, particularly within the text nodes. We therefore chose to associate a semantic label not only with the DOM element nodes but also with all comment and text nodes. This decision alleviates coarse-grained label anchors and hence facilitates an exact performance evaluation. Please note that this does not necessarily mean a lot more annotation work, as the number of semantically interesting content elements remains constant. Also, we created a tool which supports the document annotation process via a visual representation of the DOM tree. With this tool, a single webpage can generally be annotated under five minutes.

Before the actual annotation of the news pages, we specified a set of 30 labels (cf. section 3.3), 28 of which describe different items related to the principal news content. One label is used to mark DOM nodes that are part of advertisements; one represents all irrelevant DOM nodes. For all but the latter two labels, we exclusively annotate DOM leaf nodes. This is no restriction, because inner DOM nodes are not visible. For advertisements, however, we annotate complete DOM subtrees. That way advertisement removal methods can evaluate whether they were successful in pruning all relevant nodes. Finally, we had a postgraduate student label the 604 news webpages based on a label specification. A second pass over the data has been used to validate and, if necessary, correct the annotations, ensuring their uniformity.

3.3. Labels and useful statistics

The NEWS600 dataset comprises a total of 604 news webpages from 177 different domains (not counting subdomains), including practically every major news site.

In total, we count 633,664 annotated DOM nodes (element, text or comment nodes) of which 128,514 are annotated. From the 633,664 annotated nodes, 165,654 are visible leaf nodes. From those visible leaf nodes, 119,107 have no actual annotation and are henceforth associated with the label `none`. Table 2 lists the 30 different labels and their occurrences in the training and test sets. For example, the related article label marks links related to the feature story. Usually, those are indicated by keywords such as "related" and can be found beneath or next to the story. Further reading, past coverage, documents and attachments that directly further the understanding of the principal content also carry this label. Please note that some of the labels are quite rare compared to the total number of visible leaf nodes. Also, they are highly unbalanced.

4. Experiments

In this section we present first experimental results on the NEWS600 dataset. We compare both SVM and CRF and explore to what extent the set of style features contribute to the overall classification performance, analyze the results for the 30 individual classes and compare them to previous approaches ([11, 12]).

4.1. Evaluation metrics

We measure the performance of our model on the individual labels using precision, recall and F_1 -measure [10], defined as the evenly weighted harmonic mean of precision π and recall ρ : $F_1(\pi, \rho) = \frac{2\pi\rho}{\pi+\rho}$. Note that whenever there is a significant imbalance among labels, as is true for the NEWS600 dataset, true negatives tend to dominate accuracy and thus render it susceptible to mis-interpretation. In our experiments, the accuracy of all individual labels lies above 99 per cent and is hence uninformative. In Table 2 we thus do not show this metric.

In a multi-class setting there are two principal ways to combine the performance results of the different labels. Either we compute their arithmetic mean, giving equal weight to each of the labels; or we compute the mean weighting each label by the number of times they occur in the dataset (their priors), hence putting equal weight on each visible leaf node. Metrics computed the latter way are referred to as *micro-averaged*, metrics calculated using the arithmetic mean are termed *macro-averaged*. Macro-averaged metrics are often dominated by the performance on rare labels while micro-averaged metrics are dominated by the performance on frequent labels. The two ways of measuring performance are hence complementary, and both are informative.

4.2. Experimental setup

We randomly split the 604 webpages of the NEWS600 corpus into two sets, one for training (parameter estimation), containing 400 webpages, and one for testing (performance evaluation), counting 204 webpages. For each node in both sets we collect its label and features. Because we are working with sparse feature representations, only a tiny fraction of all possible features need to be stored for each node, namely the ones that are indeed present. However, the output domains of the different feature functions vary considerably in range, which is why we individually rescaled each feature by three times its standard deviation on the training set. Note that rescaling maintains sparsity and facilitates the parameter estimation process. We also remove all features that appear less than three times in the training set. This leaves us with 41,342 features out of a total of 108,692 with a sparsity of 0.24 per cent.

To measure the influence of the layout and style features on the content extraction performance, we removed all of them from the original training and test sets. The new normalized sets, termed *No Style*, contain 41,075 features with a sparsity of 0.089 per cent.

We used cross-validation on the training sets to determine reasonable parameter settings of our models. For the stochastic subgradient optimization method [8] used for the linear SVM, we found the subset size $k = 1$ and the regularization parameter $\lambda = 2 \cdot 10^{-4}$ to work well. All SVM results in this paper have been produced using 30 passes through the entire training set. For the variance of the Gaussian regularizer of the CRF [9] we used the value 1.

4.3. Results

The overall results of both the SVM and the CRF can be found in Table 1. In general, the CRF outperforms the SVM, reducing the error of the SVM by almost 20 per cent. This performance gain can be attributed to the additional transition features of the CRF (cf. section 2.2). The achieved overall accuracy of 94.21 per cent is encouraging, given that our current model is very general in what concerns features and algorithms used. Note that the values of the macro-averaged metrics are significantly lower than the micro-averaged ones, suggesting that a number of labels with few annotated nodes are often misclassified.

Although vision-based features have been shown to be beneficial under some circumstances [11], our results do not exhibit a significant increase in performance when comparing No Style and All feature sets (cf. Table 1). It thus remains to be seen to which extent and how style information can enhance the usage of structural and linguistic features.

Table 2 shows the performance results of the CRF on the individual labels for all available features. Not surprisingly, very rare labels such as author-date and contributor are hard to extract. Note that the results for author, publisher and date are all better than for author-date. It is probably both the tiny number of nodes available for the author-date label and the fact that there is a semantic overlap between the afore-mentioned labels that contribute to the low performance. On the other end, the extraction of frequent classes (advertisement, paragraph and social bookmark) can be achieved with an F_1 -measure greater than 90 per cent, which is promising. In general, however, the more precise the definition of a label, the easier it is to extract it.

In the following we analyze the learned parameters of our model and we present some of the features that turned out to be discriminative for a particular label. A paragraph, for instance, can be made out by large values of margin-bottom or by a DOM tree parent node with tag name p . A title is indicated by DOM tree parent nodes with tag names $h1$ or $h2$ or a parent class attribute that contains the string

Table 1. Experimental results for two models and two feature sets (in per cent).

Model	Feature set	Micro-averaged			Macro-averaged			Overall accuracy
		precision	recall	F ₁ -measure	precision	recall	F ₁ -measure	
SVM	No Style	92.67	92.81	92.49	85.37	55.91	64.51	92.81
	All	92.79	92.95	92.64	84.12	55.74	64.24	92.95
CRF	No Style	93.83	93.85	93.75	73.90	64.93	67.58	93.85
	All	94.12	94.21	94.11	76.16	64.12	68.03	94.21

H1; then the font size and the differences to the average and maximal document font sizes as well as font size changes (for both predecessor and successor) are relevant. The width of the title box has some discriminative power, too. A very indicative feature for the label advertisement is whether the node is within an `iframe` or not. Large distances to the average depth of `p` nodes in the DOM tree point out advertisements as well as text contents containing the words `advertisement`, `ads` or `google`. Another relevant feature is the left margin of the node's box: Is it far on the right, the node is likely to be advertisement.

Although prior work is hardly comparable with ours, mainly due to the fact that the datasets are entirely different, we nevertheless try to establish some links between similar tasks. Xue et al. [11] have a very similar approach for the extraction of titles from webpages, yet use a less restricted set of 4,258 webpages from the TREC web track .GOV data. Their best method achieves a F₁-measure of just under 80 per cent. Our experimental results (cf. Table 2) compare well, having a F₁-measure of 82.72 per cent. Although we work on a more restricted domain of webpages, we remark that their largest font size baseline is higher than ours (F₁-measure of 40 per cent), mainly due to a higher precision.

Moreover, the work of [12] is relevant to the discussion here. However, they only show results for classification accuracy. Their best method reaches an accuracy of 85.7 per cent for the binary task of main content extraction. [3] report an overall accuracy of 87.71 per cent for unsupervised extraction of main news content.

Finally, due to the rather high cost of the webpage annotation process, we also show the relation between the training set size and the test set performance of the sequential CRF (see Figure 1). Overall accuracy is averaged over 10 runs. In each run, the documents in the new training set are sampled uniformly at random from the 400 webpages in the original training set.

5. Related work

There is plenty of prior work related to this paper. What concerns the employed model (feature set and parameter estimation), Xue et al. [11] are closest to our approach. However, they are concerned with title extraction instead of news

content extraction. Contrary to our current model, they also include page segmentation results which have been obtained in a distinct pre-processing step. In evaluation, they allow for approximate title matches based on the edit distance of two strings. Ziegler et al. [12] tackle the problem of content extraction from news webpages using a particle swarm optimization on four linguistic and four structural features. With 610 webpages, their dataset is very comparable in size. Again, a similarity measure is defined to measure the discrepancy between the actual and the predicted content. Castro Reis et al. [3] cluster a set of domain-oriented webpages, generate extraction templates using tree

Table 2. Performance results (in per cent) of the CRF on individual labels using all available features. Accuracy is not shown (see discussion in section 4.1).

Label	N_{train}	N_{test}	Precision	Recall	F ₁ -measure
advertisement	19145	7479	90.06	90.13	90.10
author	263	142	67.65	64.79	66.19
author-date	13	3	0.00	0.00	0.00
caption	167	82	93.62	53.66	68.22
category	70	33	55.00	33.33	41.51
comments	165	83	48.60	62.65	54.74
contact	147	60	96.15	41.67	58.14
contributor	14	13	33.33	30.77	32.00
copyright	97	69	96.36	76.81	85.48
current page	41	21	80.00	57.14	66.67
date	404	208	80.69	90.38	85.26
e-mail article	591	296	97.90	94.59	96.22
heading	153	60	48.89	36.67	41.90
introduction	95	52	68.75	21.15	32.35
media credit	42	21	76.00	90.48	82.61
multimedia	229	123	86.54	73.17	79.30
next page	82	54	76.92	74.07	75.47
none	80399	38708	96.02	96.75	96.38
paragraph	6674	3080	89.47	94.35	91.85
print page	604	286	97.14	95.10	96.11
publisher	112	63	71.11	50.79	59.26
RSS feed	85	39	96.55	71.79	82.35
recommendation	35	17	76.19	94.12	84.21
related article	1123	621	81.97	54.91	65.77
single page	38	23	85.71	78.26	81.82
social bookmark	1379	762	94.81	93.44	94.12
subtitle	40	20	42.86	15.00	22.22
tags	193	79	69.77	75.95	72.73
title	398	204	97.99	71.57	82.72
topics	115	40	88.89	40.00	55.17

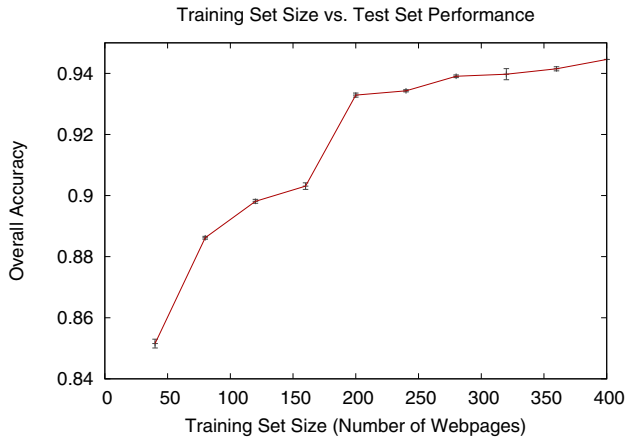


Figure 1. Overall accuracy of the sequential CRF for different training set sizes.

structure analysis for the clusters and then label the text passages that match the extracted templates. The adaptation to unseen sites is thus very likely to require the extraction of new templates. Chakrabarti et al. [2] introduce a combinatorial optimization framework for webpage segmentation based on DOM element nodes. Although concerned with segmentation rather than extraction, their approach is tightly linked with ours. To some extent, one might regard it as an instantiation of the framework we present. Further work that is related, but less relevant to this paper can be sorted by granularity into approaches that focus on the extraction of webpage blocks and those that deal with webpage segmentation. Lin et al. [7] as well as Debnath et al. [4] focus on the extraction of informative content blocks from webpages. They use heuristics based on a set of HTML tags (such as `tr`, `p` and `ul`) to partition a webpage into blocks. These blocks are then classified into informative or non-informative. Further notable examples of works in webpage segmentation which are related to our approach are Cai et al. [1] and Hattori et al. [5].

6. Conclusions and future work

In this paper we studied the problem of web content extraction. We proposed to cast this problem into a simple classification framework for which we presented two first instantiations, a linear SVM and a sequential CRF. We introduced the NEWS600 corpus, a set of 604 labeled webpages that can be used for several content extraction tasks. Our experimental results show that the CRF, which takes into account the implicit sequential structure of a webpage, clearly outperforms the SVM. Both models do not attribute significant importance to style and layout features. One future direction is hence a better integration of these features.

Acknowledgements

The first author gratefully acknowledges support from Microsoft Research through its European PhD scholarship programme. He thanks Rudy Sicard and Massih-Reza Amini for helpful feedback.

References

- [1] D. Cai, S. Yu, J.-R. Wen, and W.-Y. Ma. VIPS: a vision-based page segmentation algorithm. Technical Report MSR-TR-2003-79, Microsoft Research, Redmond, WA, USA, November 2003.
- [2] D. Chakrabarti, R. Kumar, and K. Punera. A graph-theoretic approach to webpage segmentation. In *Proceedings of the 17th International World Wide Web Conference (WWW2008)*, pages 377–386, Beijing, China, April 2008. ACM Press, New York, NY.
- [3] D. de Castro Reis, P. B. Golgher, A. S. da Silva, and A. H. F. Laender. Automatic web news extraction using tree edit distance. In *Proceedings of the 13th International World Wide Web Conference (WWW2004)*, pages 502–511, New York, NY, USA, May 2004. ACM Press, New York, NY.
- [4] S. Debnath, P. Mitra, and C. L. Giles. Automatic extraction of informative blocks from webpages. In *SAC '05: Proceedings of the 2005 ACM Symposium on Applied Computing*, pages 1722–1726, New York, NY, USA, 2005. ACM.
- [5] G. Hattori, K. Hoashi, K. Matsumoto, and F. Sugaya. Robust web page segmentation for mobile terminal using content-distances and page layout information. In *WWW '07: Proceedings of the 16th International Conference on World Wide Web*, pages 361–370, New York, NY, USA, 2007. ACM.
- [6] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289, San Francisco, CA, USA, 2001.
- [7] S.-H. Lin and J.-M. Ho. Discovering informative content blocks from web documents. In *KDD '02: Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 588–593, New York, NY, USA, 2002. ACM.
- [8] S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal Estimated sub-Gradient Solver for SVM. In *ICML '07: Proceedings of the 24th International Conference on Machine Learning*, pages 807–814, New York, NY, USA, 2007.
- [9] C. Sutton. GRMM: GRaphical Models in Mallet. <http://mallet.cs.umass.edu/grmm/>, 2006.
- [10] C. J. Van Rijsbergen. *Information Retrieval*. Department of Computer Science, University of Glasgow, 1979.
- [11] Y. Xue, Y. Hu, G. Xin, R. Song, S. Shi, Y. Cao, C.-Y. Lin, and H. Li. Web page title extraction and its application. In *Information Processing & Management*, volume 43, pages 1332–1347, January 2007.
- [12] C.-N. Ziegler and M. Skubacz. Content extraction from news pages using particle swarm optimization on linguistic and structural features. In *Web Intelligence*, pages 242–249. IEEE Computer Society, 2007.