Automatic Genre Classification in Web Pages Applied to Web Comments

Melanie Neunerdt, Michael Reyer, Rudolf Mathar
Institute for Theoretical Information Technology
RWTH Aachen University, Germany
{neunerdt, reyer, mathar}@ti.rwth-aachen.de

Abstract

Automatic Web comment detection could significantly facilitate information retrieval systems, e.g., a focused Web crawler. In this paper, we propose a text genre classifier for Web text segments as intermediate step for Web comment detection in Web pages. Different feature types and classifiers are analyzed for this purpose. We compare the two-level approach to state-of-the-art techniques operating on the whole Web page text and show that accuracy can be improved significantly. Finally, we illustrate the applicability for information retrieval systems by evaluating our approach on Web pages achieved by a Web crawler. [1]

1 Introduction

The high amount of social media tools lead to constantly growing user generated content in the Web. Different types of Web sites, e.g., blogs, forums as well as news sites provide functionalities to post Web comments to related articles. Whereas an abundance of Web comments is publicly available, the size of the World Wide Web makes it a challenging task to identify Web pages with Web comments. One solution to build topic specific Web comment corpora, is a focused crawler jointly using a topic classifier and a Web comment classifier. Altogether, automatic Web comment detection has the potential to significantly improve the performance of information retrieval systems and provide corpora, which can efficiently be used, e.g., for marketing studies.

In this paper, we use *Web comment* as a particular text genre. It is fundamental to consider the fact, that a Web page can be a composition of different text genres. For example Web comments posted to a particular article on the same Web page obviously comprise the text genre *article* and *Web comment*. It is to be expected that the associated feature vectors of different text genres show different characteristics which can be identified more easily if they are investigated separately. Therefore, we apply a two-level classification approach. We first classify the text genre of each text segment of the Web page. On the second level we declare a Web page to be relevant, if a Web comment is detected in at least one text segment.

The outline of this paper is as follows. After reviewing related work in Section 2, we propose the two-level classification approach and discuss potential features for detecting Web comments in Web pages in Section 3. Section 4 introduces the corpora used and Section 5 reports experimental results. Finally, we conclude our work and discuss future research directions.

2 Related Work

State-of-the art Web text genre classification approaches differ mainly in the feature set they use and the genre classes they define. Various types of features have been proposed for automatic text genre classification. Web pages are additionally

equipped with information such as, formatting information from HTML tags or css-style classes, and meta information given in the URL of a Web page, which further increases the possible feature dimension. Classes for Web genre classification are often related to the type of Web page. They lead from very few classes, e.g., seven in (Lee and Myaeng, 2002), to fine-grained classification with fifteen genres, (Lim et al., 2005). In (Meyer zu Eissen and Stein, 2004) a user study on Web genre class usefulness is performed. As a matter of fact, selected genres influence feature selection for automatic classification.

(Meyer zu Eissen and Stein, 2004; Lim et al., 2005; Qi and Davison, 2009) propose to use style-related HTML features, e.g., HTML tag frequencies, token-related features, e.g., text statistics or digit frequencies and POS features, e.g., noun or verb frequencies. In (Lim et al., 2005) some URL-related features, e.g., depth of URL, are proposed as additional features. (Qi and Davison, 2009) particularly investigate Web-specific features and their usability for different Web page classification tasks, e.g., sentiment classification and subject classification. Beside, on-page features, directly located on the page to be classified, they investigate the usability of features of linked pages. The performance of different POS-related features is particularly studied in (Feldman et al., 2009; Santini, 2004). (Feldman et al., 2009) propose to use POS histograms over a sliding window as features. Compared to the results achieved by a classifier working with POS trigram features a significant performance increase is reported. All these approaches achieve classification accuracies between 70% and 80%.

3 Classification Approach

The goal of our work is to detect Web comments in Web pages. The overall approach is simple, Web pages are split into small text segments, which are represented by feature vectors and thereby classified. As we aim at showing that the classification can significantly be improved by the segmentation approach, we apply a very simple rule based segmentation based on HTML tags: All tags, but links <*a*>, line breaks <*br*> and some font tags, e.g., <*small*> and <*strong*>, are used for splitting. The result is a very fine-grained segmentation, which might result in splitted Web comments or articles. We believe that splitting in too many segments will not effect the performance too much. Though, the parametrization of the classification may be effected, e.g., k.

In the following we mathematically describe the two-level classification problem, classifying Web text segments on first level and Web pages on second level. We define the index set $\mathcal{PM} = \{1, \ldots, P\}$ for Web pages. Each Web page p is splitted into a sequence of N_p text segments,

$$\mathbf{S}^p = \left(\mathbf{s}_1^p, \ldots, \mathbf{s}_{N_p}^p\right),$$

where each text segment $\mathbf{s}_i^p$, $i = 1, \ldots, N_p$ is represented by an n-dimensional feature vector

$$(\mathbf{s}_i^p)^T \in \mathcal{X} = \mathcal{S}_1 \times \mathcal{S}_2 \times \ldots \mathcal{S}_n.$$

The aim of the segment classification is to predict the to $\mathbf{S}^p$ associated genre class vector

$$\mathbf{c}^p = \left(c_1^p, \ldots, c_{Np}^p\right),$$

with $c_i^p \in \mathcal{C}$ for $i = 1, \ldots, N_p$ and $\mathcal{C}$ comprises the set of genre classes.

First of all, we consider a seven-class problem. We differentiate between the seven classes, $WebCOMment$, $ARTicle$, $USEr$, $TITle$, $TIMe$, $METa$ and $OTHer$, represented by

$$\mathcal{C} = \{COM, ART, USE, TIT, TIM, MET, OTH\}. \quad (1)$$

Applying this approach, related information like the posting date, the user name or the related article are automatically identified. The Web comment corpus quality benefits from such information. However, for Web comment detection the binary decision if the class is $COMment$ or not is sufficient. In order to solve the sequence labeling task, the optimization problem

$$\hat{\mathbf{c}}^p = \arg\max_{\mathbf{c}^p} \left\{g\left(\mathbf{S}^p, \mathbf{c}^p\right)\right\}$$

where g represents any decision function, is solved. This is a huge optimization problem, which is simplified by two assumptions. First, the genre classes $\hat{c}_i^p$ are predicted independently from predictions $\hat{c}_j^p$ for $j \neq i$ by

$$\hat{c}_i^p = \arg\max_{c_i^p} \left\{g\left(\mathbf{S}^p, c_i^p\right)\right\}.$$

Second, we assume that the text genre class for a given text segment $\mathbf{s}_i^p$ at position i only depends on some - here k - preceeding and succeeding text segments. Hence, the optimization problem is reformulated as

$$\hat{c}_i^p = \arg\max_{c_i^p} \left\{g\left(\mathbf{s}_{i-k}^p, \ldots, \mathbf{s}_i^p, \ldots, \mathbf{s}_{i+k}^p, c_i^p\right)\right\}.$$

Note that, for the first and last segments of each Web page p, there are not enough predecessor and successor segments available. In such cases the number of considered predecessor and successor segments is reduced to the maximal possible amount. Web pages are considered to be *RELevant*, if at least one segment is classified as $COMment$. Hence, the condition for the second level classification is

$$\sum_{i=1}^{Np} 1_{\{COM\}}\left(\hat{c}_i^p\right) \geq 1, \text{ with } 1_{\mathcal{A}}(x) = \begin{cases} 1, & x \in \mathcal{A} \\ 0, & \text{else} \end{cases}$$

representing the indicator function for any set $\mathcal{A}$.

In order to compare our approach to existing approaches, we directly solve the classification on Web page level. We represent the whole Web page text by the same feature vector than for later Web page text segments. Hence, we solve the optimization problem for $N_p = 1$.

3.1 Web Page Feature Types

We generally expect that the combination of several features from each level can be used to identify text segments as elements of $\mathcal{C}$, which can then be used to identify relevant Web pages. In our approach we combine some of the features proposed in (Lim et al., 2005; Meyer zu Eissen and Stein, 2004; Kohlschütter et al., 2010) with some new features, which results in 102 features in total. New features are introduced for all three feature types. Such features are motivated by an extensive study of the language in Web comments and the structure of Web pages like blogs and forums.

3.1.1 Token-based Features

Token-based features are easily accessible, without any text preprocessing. However, in order to develop a topic independent solution, token-level features need to be carefully selected. We extend simple frequency count features, e.g., punctuation marks, digits or symbols, by Web comment related features. Emoticons, letter iterations, e.g., Hallllloooo (Hellllloooo), multiple punctuations, e.g., !!!, ?!, @ symbols, uncapitalized words, etc. are taken as additional features counting the frequency. Furthermore, some sentiment related features are defined. Adjectives in Web comments are inherently connected with evaluative judgements. Hence, frequency counts of positive and negative orientated adjectives are promising features for differentiating Web comments from other texts. The *SentiWS* word list proposed in (Remus et al., 2010) is used for such frequency counts.

Finally, we complement some features proposed in previous works. (Kohlschütter et al., 2010) propose a text density measure particularly for Web text segment classification. From, e.g., (Lim et al., 2005) we take over frequency counts of *content, function* and *unusual* words. In total 50 token-based features are used for the classification.

3.1.2 POS-based Features

Many approaches introduce features based on Part-of-Speech (POS) information for text classification. Basically, such features are simple frequency counts of single POS tags (1-gram) or ratios between different POS tags, e.g., the verb-noun ratio. For our approach we combine POS-based features proposed in (Lim et al., 2005; Meyer zu Eissen and Stein, 2004) using the STTS tagset (Schiller et al., 1999) with 54 part-of-speech tags. As a tagger we use WebTagger (Neunerdt et al., 2014) particularly developed for social media texts. Previous studies have shown, that due to the dialogic style of Web comments, particularly the sequence of POS tags are different. E.g., in (Neunerdt et al., 2013) POS trigram (3-gram) statistics evaluated on a social media text corpus show significant differences compared to newspaper texts. Hence, POS 3-grams seem to be a good feature to differentiate Web comments from other texts. However, to determine reliable POS tag features requires automatic POS tagging with high accuracies. Common state-of-the art taggers achieve high accuracies on newspaper texts, which significantly drops when applied to unstandardized texts, such as Web comments. For the main classification approach we use 38 POS-based features. Note that, POS-based features, such as sentence length statistics, are also included here, since we use POS tags to detect the end of a sentence.

3.1.3 HTML-based Features

Structural features, based on HTML tags (headline *<h1>*, paragraph *<p>*, etc.) or CSS

classes are commonly used for Web page classification approaches. Unfortunately, the usage of CSS classes and HTML tags are mainly Web site specific. The increasing usage of CSS classes and styles makes it even more difficult to infer semantic relations between HTML tags and text segments. However, CSS class names are not chosen arbitrarily and often have a semantic relation to the text elements they are defined for, e.g. the user, the date or the web comment. This allows us to define useful features based on such CSS class names. For example the style of a Web comment is frequently defined by CSS class names like, e.g., *comment, post, message*. Based on a list of common class name strings, we define binary features marking, if one of the strings is contained in the CSS class name of the current segment. In addition, we define another binary feature, marking the presence of HTML tags, which never jointly occur with Web comments due to their functionality, e.g., *h1 option, title, em, button*. Finally, we use further structure related features. Since, the position of the Web segment is often a good hint for the corresponding class, we introduce that as additional feature. For example, Web comments are often located below the article at the end of a Web page. Some other features, e.g., the link density, are taken from (Meyer zu Eissen and Stein, 2004). In total, 14 HTML-based features are used for classification.

4 Corpora

Evaluations are performed on two different corpora, a manually collected Web page corpus for training and testing, and a collection of Web pages accessed with a crawler for validation. Both corpora are selections of German Web pages solely.

4.1 Web Comment Collection

The Web comment collection is created particularly to train a Web comment classifier. It consist of 336 manually assessed Web pages from 237 different Web sites/domains. 71% of the Web pages contain at least one posted Web comment. The remaining 99 Web pages contain Web comments related articles. The Web pages contain forums, blogs and different news sites dealing with different topics. In this paper we call that corpus *Web Comment Train* (*WCTrain*). First we apply the segmentation described in Section 3 to each Web page. Considering the visual representation in a Web browser, plain Web text segments are labeled by four human annotators as either $WebCOMment$, $ARTicle$, $USEr$, $TITle$, $TIMe$ or $METa$ (text, which gives any further meta information to another text/author). Note that, every page is labeled by one annotator, since we do not expect significant inter-annotator disagreement in this context. Unselected text is regarded to $OTHer$ (no content, left over class). The distribution of all classes at token (including non-words)-, word- and segment-level is depicted in Table 1.

Class	# Segments	# Words	# Tokens
Total	45,955	479,483	596,630
$WebCOMment$	5.36 %	36.78%	35.10%
$TITle$	2.94%	1.53%	1.52%
$TIMe$	4.79%	0.62%	1.01%
$METa$	2.16%	0.71%	1.59%
$USEr$	4.43%	0.83%	0.82%
$ARTicle$	1.59%	25.34%	23.89 %
$OTHer$	78.74%	34.19%	36.07%
$NonCOMment$	94.64%	63.22%	64.90%

Table 1: Class distribution in the *WCTrain* Corpus.

4.2 Crawl Collection

A second corpus is introduced, with the goal to evaluate the applicability of the developed Web comment classifiers for information retrieval systems. The Web page corpus is acquired by starting a crawl process from 112 seed pages. We manually have selected the seed pages from 78 different domains, fulfilling one of the two criteria: The Web page is a blog, forum or news site, which contains at least one Web comment. The Web page is a so called hub page, which contains a high number of links to Web pages, which also fulfill the first requirement. The crawl process results in 72,534 Web pages from 1414 different Web domains. For the sample corpus *Web Page Crawl* (*WPCrawl*) 827 Web pages are selected randomly from the basic crawl result. In contrast to the *WCTrain* corpus the annotation is performed on Web page level rather then Web segment level. Two human annotators label each Web page as Web pages are labeled by four human annotators as *RELevant*, if it contains at least one Web comment. All remaining Web pages are regarded to be *non-RELevant*. In total, 57% of such Web pages are *RELevant*. This corpus serves as validation for classification on the second level.

5 Experimental Results

In this section we analyze different classifiers utilizing different feature combinations. In order to build a good information retrieval system, it is important to achieve high precision rates for the particular $COMment$ class. High precision means, high quality Web comment corpora. Therefore, we particularly study the classifiers, considering precision rates P_{COM} on segment level for the COM class. For our experiments we use the WEKA software, (Hall et al., 2009). We analyze three different classifiers, a KNN classifier, a decision tree (J48) and a Support Vector Machine (SVM). For the KNN classifier we used the weighted Manhattan distance as a metric considering $K = 9$ next neighbors, which gave the best result for $K = 1 \ldots 15$. Varying the decision tree threshold of the minimum number of objects in a leaf from 2 to 15 and choosing between binary and non-binary split, we used a non-binary tree with a threshold of 6, which gave the best result. For the SVM classifier a Pearson VII function-based universal kernel achieves best results.

5.1 Validation on *WCTrain* Corpus

Cross validation results for the three classifiers using an n-dimensional feature vector are depicted in Table 2. We measure classification accuracy by COM class precision P_{COM}, COM class recall R_{COM}, average F_1-Score, average ROC Area under Curve (AuC) and total accuracy (AC). The upper part of Table 2 depicts classification accuracies achieved with the three different classifiers using all proposed features. In order to analyze the influence of integrating features from predecessor and successor segments for classification in more detail, classification accuracies for different values of parameter k are depicted in addition.

Comparison of the classifiers for $k = 0$ shows not much difference in total accuracies (AC). However, considering P_{COM} class precision results, the KNN classifier significantly outperforms the other approaches. Highest precision rates for the 2-class and 7-class are achieved for $k = 2$ for all classifiers. This confirms our assumption that similar small sequences of text segments occur in Web pages. Hence, considering the text segments close by are useful features for text genre prediction. The KNN classifier solving a 2-class problem achieves the highest precision rate with 0.94. Hence, beside using different classifiers, the values of k allow for further adjustments towards P_{COM} precision rates without decrease of total accuracies.

Considering that KNN achieves the best results for $k = 0$, we exemplarily investigate the performance of KNN classifier, for each feature type separately. Results are depicted in the middle part of Table 2. Using an approach based on token-based features outperforms the POS-based and HTML-based approach. However, classification accuracy drops significantly, compared to the approach, when using all feature types in combination (KNN (k=0)). We further investigate different feature types by calculating the per-feature information gain. Figure 1 shows the features in decreasing order of their information gain for the 2-class and 7-class problem. Information gain values are below 0.17 for the 2-class and below 0.35 for the 7-class problem. Generally, simple token-based features, e.g., capitalized token counts or letter counts, and POS-based features, e.g., POS tags counts or verb noun-ratio, appear to be strong indicators for class membership for the 2- and 7-class problem. Confirming the results achieved with the classifier using HTML-based features solely, HTML-based features are lower ranked. However, assessing features usability by the information gain rates them independently. In order to investigate the combination of different feature considering their redundancy, we apply a greedy correlation-based feature subset selection proposed by (Hall, 1998). Subsets of features that are highly correlated with the class while having low intercorrelation are preferred. Results are depicted in the lower part of Table 2. P_{COM} rates are slightly lower, compared to the classifiers using all 102 features, however the number of features can significantly be reduced by 4/5, which reduces computational classification effort. Analyzing the resulting feature subsets, e.g., for the 7-class problem results in a combination of 36% token-based, 41% POS-based and 23% HTML-based features. We conclude that, the selected subsets of different feature types as well as relatively low per-feature information gain but at the same time high acceptable classification accuracy shows that particularly combining features from

Algorithm	n	P_{COM}		R_{COM}		Average F_1-Score		Average AuC		AC	
		2-class	7-class	2-class	7-class	2-class	7-class	2-class	7-class	2-class	7-class
KNN (k=0)	102	0.88	0.87	0.70	0.73	0.88	0.76	**0.99**	0.97	0.978	0.915
KNN (k=1)	306	0.93	0.91	0.79	0.82	0.92	**0.81**	**0.99**	**0.98**	0.985	0.934
KNN (k=2)	510	**0.94**	**0.93**	0.85	**0.86**	**0.94**	**0.81**	**0.99**	**0.98**	**0.988**	0.938
SVM (k=0)	102	0.87	0.84	0.75	0.80	0.90	0.74	0.87	0.92	0.980	0.911
SVM (k=1)	306	0.88	0.89	0.86	0.80	0.93	0.77	0.93	0.92	0.986	0.926
SVM (k=2)	510	0.90	0.90	**0.88**	0.78	**0.94**	0.69	0.94	0.90	**0.988**	0.908
J48 Tree (k=0)	102	0.80	0.78	0.73	0.73	0.88	0.75	0.93	0.91	0.976	0.912
J48 Tree (k=1)	306	0.82	0.79	0.75	0.75	0.89	0.76	0.93	0.91	0.978	0.935
J48 Tree (k=2)	510	0.83	0.79	0.76	0.78	0.89	0.76	0.93	0.89	0.979	**0.958**
KNN, token features	50	0.83	0.80	0.63	0.67	0.85	0.63	0.98	0.93	0.973	0.878
KNN, POS features	38	0.80	0.78	0.62	0.65	0.85	0.61	0.96	0.92	0.971	0.865
KNN, HTML features	14	0.71	0.64	0.48	0.55	0.78	0.59	0.94	0.90	0.962	0.855
KNN, Subset features	23/22	0.87	0.82	0.62	0.69	0.86	0.60	0.95	0.91	0.975	0.872
SVM, Subset features	23/22	0.80	0.76	0.39	0.70	0.75	0.50	0.69	0.84	0.962	0.863
J48, Subset features	23/22	0.78	0.74	0.67	0.71	0.86	0.62	0.94	0.90	0.973	0.878

Table 2: Cross Validation Results achieved for different classification approaches 2-class/7-class.

2-class problem									
	KNN Classifier			Decision Tree			SVM Classifier		
k	P_{REL}	R_{REL}	AC_{PAGE}	P_{REL}	R_{REL}	AC_{PAGE}	P_{REL}	R_{REL}	AC_{PAGE}
0	0.83	0.83	0.80	0.68	**0.98**	0.72	0.84	0.94	**0.86**
1	0.83	0.86	0.82	0.75	0.96	0.79	0.73	0.95	0.77
2	**0.87**	0.85	0.84	0.68	0.92	0.71	0.83	0.94	**0.86**
Classification results achieved on total Web page with $N_p = 1$									
	0.78	0.85	0.78	0.73	0.78	0.71	0.75	0.88	0.76

Table 3: Validation on *WPCrawl* corpus.

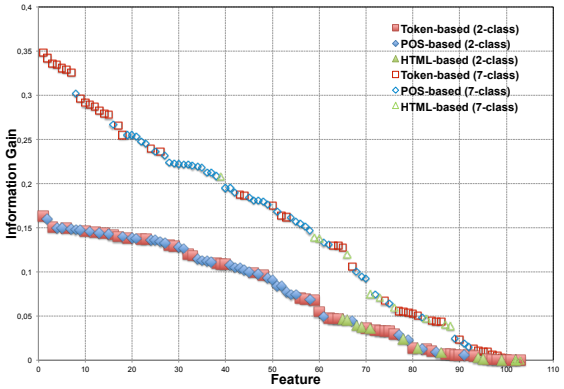

Figure 1: Information gain of different feature types applied to 2- and 7-class problem.

different types is particularly important.

5.2 Validation on *WPCrawl Corpus*

In order to show the usability of our classification approach for information retrieval tasks, we apply our classifier on the *WPCrawl Corpus*. Results are depicted in Table 3. Precision P_{REL}, recall R_{REL} and total accuracy AC_{PAGE} are given on the second classification level for a total Web page rather than a text segment. Hence, e.g., P_{REL} is the number of $RELevant$ Web pages classified as $RELevant$ divided by the total number of $RELevant$ pages. Considering the task of building a Web comment corpus by selecting all $RELevant$ classified pages, high P_{REL} are particularly important. However, the resulting corpus size is even important and hence R_{REL} is not neglectable. Best P_{REL} results are achieved with the KNN classifier with $k = 2$. In this case 463 Web pages would be selected from the original *WPCrawl* corpus, where 87% would be $RELevant$ pages. For comparison, the last column shows results achieved with the classifiers performed without segmentation, on the whole Web page. The highest P_{REL} of 0.78 is achieved with the KNN classifier, which is significantly lower compared to our two-level classification.

6 Conclusion

In this paper, we presented a simple approach for Web comment detection classifying Web text segments as intermediate step. The two level classification particularly improves precision rates compared to a classifier applied to the whole Web page text. Applying our classifier combining token, POS and HTML-based for Web comment corpus refinement to Web pages accessed by a crawler, shows significant improvement in corpus quality. The amount of relevant Web pages containing Web comments, could be improved from 57% to 87% using a KNN classifier.

The presented results raise research in many different directions. Results achieved by feature extensions, motivate to use a Markov model classifier to label Web text sequences. That would allow to model dependencies of predecessor classification results and could further improve classification accuracies. Furthermore, we need to investigate possibilities for feature selection in more detail, to reduce the complexity of the classifier.

References

Sergey Feldman, Marius A. Marin, Mari Ostendorf, and Maya R. Gupta. 2009. Part-of-speech Histograms for Genre Classification of Text. In *ICASSP*, pages 4781–4784. IEEE.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November.

M. A. Hall. 1998. *Correlation-based Feature Subset Selection for Machine Learning*. Ph.D. thesis, University of Waikato, Hamilton, New Zealand.

Christian Kohlschütter, Peter Fankhauser, and Wolfgang Nejdl. 2010. Boilerplate Detection Using Shallow Text Features. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, pages 441–450, New York, NY, USA. ACM.

Yong-Bae Lee and Sung Hyon Myaeng. 2002. Text Genre Classification with Genre-revealing and Subject-revealing Features. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 145–150, New York, NY, USA. ACM.

Chul Su Lim, Kong Joo Lee, and Gil Chang Kim. 2005. Multiple Sets of Features for Automatic Genre Classification of Web Documents. *Inf. Process. Manage.*, 41(5):1263–1276.

Sven Meyer zu Eissen and Benno Stein. 2004. Genre Classification of Web Pages: User Study and Feasibility Analysis. In *KI 2004: Advances in Artificial Intelligence*, pages 256–269. Springer Berlin Heidelberg.

Melanie Neunerdt, Michael Reyer, and Rudolf Mathar. 2013. Part-of-Speech Tagging for Social Media Texts. In *Proceedings of The International Conference of the German Society for Computational Linguistics and Language Technology*.

Melanie Neunerdt, Michael Reyer, and Rudolf Mathar. 2014. Efficient Training Data Enrichment and Unknown Token Handling for POS Tagging of Non-standardized Texts. In *Proceedings of KONVENS 2014*. In Press.

Xiaoguang Qi and Brian D. Davison. 2009. Web Page Classification: Features and Algorithms. *ACM Comput. Surv.*, 41(2):12:1–12:31.

Robert Remus, Uwe Quasthoff, and Gerhard Heyer. 2010. Sentiws – a publicly available german-language resource for sentiment analysis. In *Proceedings of the 7th International Language Resources and Evaluation (LREC'10)*, pages 1168–1171.

Marina Santini. 2004. A shallow approach to syntactic feature extraction for genre classification, cluk 7: The uk special-interest group for computational linguistics. In *Proceedings of the 7th Annual Colloquium for the UK Special Interest Group for Computational Linguistics*.

Anne Schiller, Simone Teufel, Christine Stöckert, and Christine Thielen. 1999. Guidelines für das Tagging deutscher Textcorpora mit STTS. University of Stuttgart.