FlawFinder: A Modular System for Predicting Quality Flaws in Wikipedia

Notebook for PAN at CLEF 2012

Oliver Ferschke[‡], Iryna Gurevych[†‡], and Marc Rittberger[†]

† Information Center for Education
German Institute for Educational Research and Educational Information

‡ Ubiquitous Knowledge Processing Lab
Department of Computer Science
Technische Universität Darmstadt

http://www.ukp.tu-darmstadt.de

Abstract With over 23 million articles in 285 languages, Wikipedia is the largest free knowledge base on the web. Due to its open nature, everybody is allowed to access and edit the contents of this huge encyclopedia. As a downside of this open access policy, quality assessment of the content becomes a critical issue and is hardly manageable without computational assistance. In this paper, we present FlawFinder, a modular system for automatically predicting quality flaws in unseen Wikipedia articles. It competed in the inaugural edition of the Quality Flaw Prediction Task at the PAN Challenge 2012 and achieved the best precision of all systems and the second place in terms of recall and F_1-score.

1 Introduction

On July 13th, 2012, the English Wikipedia reached its four millionth article[1]. Since the launch in 2001, the growing community around the free online encyclopedia has produced a knowledge base that is unprecedented in its size and coverage, which is not least due to the policy that *anyone can edit* its content. In contrast to Wikipedia's unsuccessful predecessor, Nupedia, which allowed only experts to contribute in order to maintain a high quality standard, Wikipedia's open policy was the main engine of the project's success. However, as a downside of this approach, there is no traditional editorial board that ensures the information and text quality of the articles. Hence, quality assurance is reduced to the *many eyes principle*, hoping that many editors with equal rights monitor each other and thus produce a high quality output. While this approach might work on a global level – the overall quality of Wikipedia has found to be on par with the quality of major printed encyclopedias [12] – the quality of individual articles on different levels of maturity cannot be ensured this way. Since Wikipedia is work in progress, there needs to be a way to determine the flaws of any given article in order to provide readers with an estimation of the article's quality level and give authors a

[1] http://blog.wikimedia.org/2012/07/13/english-wikipedia-crosses-4-million-article-milestone/

Flaw	Description	Training	Test
Advert	The article appears to be written like an advertisement and should be rewritten from a neutral point of view.	1 109	2 000
Empty section	The article has at least one section that is empty.	5 757	2 000
No footnotes	The article includes a list of references, related reading or external links, but its sources remain unclear because it lacks inline citations.	3 150	2 000
Notability	The article does not meet the general notability guideline.	6 068	2 000
Original research	The article may contain original research and should be improved by verifying the claims made and adding references.	507	1 014
Orphan	The article is an orphan, as no other articles link to it.	21 356	2 000
Primary sources	The article relies on references to primary sources or sources affiliated with the subject and does not contain sufficient citations from reliable and independent sources.	3 682	2 000
Refimprove	The article needs additional citations for verification.	23 144	1 998
Unreferenced	The article does not cite any references or sources.	37 572	2 000
Wikify	The article needs to be wikified, i.e. internal and external links should be added.	1 771	1 998
Untagged	Article without any cleanup templates.	50 000	–

Table 1: Flaw definitions and numbers of training and test instances per flaw. The training sets exclusively contain articles tagged with the respective flaw (except for *untagged*). The test sets contain a balanced number of flawed and untagged articles.

guide leading them to the most pressing problems in the encyclopedia. In this paper, we present FlawFinder, a modular system for automatically predicting quality flaws in unseen Wikipedia articles. It competed in the inaugural edition of the Quality Flaw Prediction Task at the PAN Challenge 2012 and achieved the best precision of all systems and the second place in terms of recall and F_1-score.

2 Task Definition

As an integral part of Wikipedia's quality assurance process, authors and articles maintainers use cleanup templates to mark articles that do not meet Wikipedia's quality requirements. Adding such a template posts an info message to the article in order to make readers aware of the existing problems. The article is furthermore added to the respective cleanup category in order to foster article maintenance. Anderka et al. [1] provide a comprehensive breakdown of the cleanup templates in the English Wikipedia. In this task of the PAN Challenge, cleanup templates are understood as indicators for quality flaws. Given a sample of articles that have been tagged with a cleanup template t thus marking it with a quality flaw f, it has to be decided whether or not an unseen article suffers from f. The task targets the prediction of ten important quality flaws for which the organizers provide a training and a test corpus. The training corpus consists of 154,116 articles extracted from the English Wikipedia snapshot from January 4th, 2012[2] which are labeled with the respective quality flaws. The test corpus contains a balanced number of flawed and untagged articles and has a total size of 19,019 documents. Table 1 shows the flaw definitions as they are stated on the template information pages and lists the numbers of articles for each flaw in the training and the test corpus.

[2] `http://dumps.wikimedia.org/enwiki/20120104`

3 Related Work

Quality assessment is a complex issue, since information quality (IQ) is a multidimensional concept that cannot be captured by a single, universal model. Many IQ models have been developed to fit the individual needs of different types of data, applications and users. Successful quality assurance is a particularly challenging issue in open, collaborative environments, since regulatory authorities are less pronounced than in conventional work environments and the concept of quality that the individual collaborators have differs greatly across the community.

From an information scientific perspective, Stvilia et al. [22] developed a framework for information quality assessment and measurement that is supposed to serve as a guide for developing information quality measurement models for many different settings. The authors systematically identified 22 information quality dimensions in the categories *intrinsic quality*, *relational and contextual quality*, and *reputational quality*. They furthermore provide metrics for automatically or semi-automatically measuring the quality along nine of the 22 dimensions. In later work [23], the authors analyzed the organization of information quality assurance work in Wikipedia by analyzing 60 discussion pages in order to identify the types of IQ problems that are most discussed by the community along with related causal factors. Building on this, Wu et al. [26] developed a framework for automatic quality assessment based on a set of 28 metrics.

Yaari et al. [27] performed a user study asking 64 people to assess the quality of five articles from the Hebrew Wikipedia in order to find the criteria which assist users in determining that an article is of high or low quality. Hereby, the authors rely both on the article page and the article revision history. They found 21 criteria, which they divide into measurable criteria, such as article length or number of edits, and non-measurable criteria, such as coverage or writing style. They furthermore analyze the discriminativeness of each criterion for articles rated as high quality and articles of low quality.

A growing body of work addresses automatic quality assessment in Wikipedia. The majority of these works targets the prediction of community created quality labels that identify *good articles* (GA) and *featured articles* (FA) (i.e. very good articles) [24,17,16]. In contrast to other labels, which can be assigned by any user, articles have to be nominated and reviewed in order to obtain GA or FA status. In other works, the *WikiProjects article quality grading scheme*[3] has been used as an additional set of gold standard labels [15,21,5,14,13].

The major problem with approaches using these community created quality labels as a gold standard for quality assessment is the small fraction of articles that are marked with these quality tags. As of August 2012, only 0.089% of all articles in the English Wikipedia are marked as *featured* and 0.39% are marked as *good*. In turn, the WikiProject quality classification only applies to selected topics and has different assessment criteria depending on the individual WikiProject[4].

In order to shed light on the quality and problems of articles *not* tagged with quality labels, Anderka et al. thus tackle the task from a different direction and use cleanup templates as indicators for quality flaws. In [1], they provide a breakdown of quality

[3] http://en.wikipedia.org/wiki/WP:ASSESS\#Grades

[4] http://en.wikipedia.org/wiki/WP:PROJ

flaws in the English Wikipedia, while they evaluate their effectiveness for quality assurance in [2] by analyzing the evolution of quality flaw markers over time. Finally, in [3], the authors automatically identify quality flaws by predicting the cleanup templates in unseen articles, which is also the goal of the Quality Flaw Detection task in the PAN challenge.

4 System Architecture

FlawFinder has been implemented as a modular and highly flexible system based on the Unstructured Information Management Architecture (UIMA) [8]. UIMA enables applications to be decomposed into reusable components which can be freely arranged into processing pipelines. We use Natural Language Processing components from the open-source NLP toolkit DKPro[5], which provides solutions for many recurring tasks like tokenization or sentence splitting and offers UIMA integration for state-of-the-art NLP components such as the Stanford CoreNLP library. As a runtime environment, FlawFinder uses the DKPro Lab [4], a lightweight framework that allows to combine independent NLP pipelines into one integrated and highly configurable system.

FlawFinder consists of five components, a corpus reader, a preprocessor, a feature extraction unit, a module for training and classification, and a report writer.

Rather than reading the provided training data directly, the corpus reader accesses the articles from our own Wikipedia database, since it offers a wider range of meta data such as the article link structure and information about the article revision history. The database has been created from the same Wikipedia data dump as the provided training corpus and the cleanup templates which are supposed to be predicted are removed. Access to the data is achieved using JWPL [28], a database driven open-source API for accessing Wikipedia, and the Wikipedia Revision Toolkit [10], a package that provides easy access to the article revision history. The corpus reader identifies the relevant articles using the page ids that are provided in the training corpus.

The preprocessor module uses DKPro components for sentence splitting, tokenization, stop word annotation and named entity recognition. Other DKPro components can easily be added to the pipeline. For parsing the MediaWiki markup of the Wikipedia articles, we use the SWEBLE MediaWiki parser [6,7], which produces an object model of the article structure.

The feature extraction component has been implemented using ClearTK [20], a UIMA-based framework for developing statistical NLP components. It offers interfaces for creating feature extractors that can be used independently from the utilized machine learning algorithm. This enables to create a highly configurable feature extraction pipeline that does not restrict the downstream components for training and classification.

5 Features

With our feature set, we aim at modeling the aspects of the article that are most likely to indicate the presence or absence of a quality flaw. Overall, we extract 31 feature types

[5] `http://code.google.com/p/dkpro-core-asl/`

which can be subdivided into seven categories described in this section. A systematic overview can be found in Table 2.

Structural Features are supposed to capture basic structural properties and surface features of the Wikipedia articles. We use the SWEBLE parser for parsing the Wiki markup and create a Wikitext Object Model (WOM) representation of the article. From this WOM, we extract all article sections along with their headers. We use the number of sections, the mean length of the section texts and the number of empty sections as features. Furthermore, we extract a plain text representation without Wiki markup from the WOM and calculate the ratio of markup to plain text as a fourth structural feature.

Reference Features capture aspects regarding the use of citations in the article. There are basically two types of references, *footnote style* references and *bibliography style* references. Footnote style references are marked with `<ref>` ... `<\ref>` tags directly within the text and are automatically listed at the bottom of the page[6]. Bibliography style references are manually listed at the end of the article, usually in the *References* section. They can either be created as manually formatted list items or can be marked with `cite` or `citation` tags for automatic reference formatting. First, we check whether manually created bibliography items exist in the *References* section and how many elements it contains. Then we count the number of all inline references in the article and determine their average number per sentence. Finally, we determine the ratio of the number of all references to the length of the article. Analogously to lists of references, it is possible to define lists of explanatory notes using the {{`notelist`}} template. It is usually placed in the *Notes* section and gathers all occurrences of explanatory notes which are defined within the text with `efn` templates. We extract this information in the same way as the references.

Network Features reflect the connections of an article within the whole network of Wikipedia articles and to external resources. Since the number of inbound links (i.e. the number of times other articles link to a given article) cannot be determined by parsing the articles in the provided corpora alone, we use the respective information from our JWPL Wikipedia database. When creating a new Wikipedia database from a Wikipedia data dump, JWPL automatically parses the articles using the JWPL Wikitext parser and stores the link information in the database. For each article, we determine the number of wiki-internal inbound links, wiki-internal outbound links and links to resources outside of Wikipedia.

Named Entity Features capture the number of named entities in the article. We use the Stanford Named Entity Recognizer [11] using the 3-class model with distributional similarity features[7] for tagging all entities of the types Person, Organization and Location. We use both the overall named entity counts and the average number of named entities per sentence as features.

[6] Depending on the setup of the page, the references might appear in different sections such as *References*, *Notes* or *Citations*.

[7] `http://nlp.stanford.edu/software/CRF-NER.shtml`

Revision-based Features are based on meta data derived from the article revision history. We use the Wikipedia Revision Toolkit [10] to determine the number of revisions for each article. Furthermore, we count the number of unique users that edited the page in the past. Since this number also includes anonymous users, which might be counted several times due to changing IP addresses, we additionally determine the number of unique registered users. Finally, we capture the age of the article in days.

Lexical Features are extracted from the plain article text that we obtain from the WOM. Any Wiki markup is removed with the exception of internal and external links. All links are replaced with a generic `EXPLICITLINK` label. Furthermore, we perform stopword filtering using the stopword list from the snowball stemmer[8], which we augmented with punctuation marks. We extract all token-unigram, bigrams and trigrams from each article and disregard any ngrams with a frequency lower than 5 across the corpus. This cutoff value was determined empirically during the parameter optimization run. We found that a value of 5 was optimal for all flaws.

Other Features include character counts, token counts and sentence counts per article. Furthermore, we measure the discussion activity by means of counting the number of individual discussion topics on the Talk page associated with the article. According to Ferschke et al. [9], we regard each titled section on the Talk page as an individual topic. We refrain from using lexical features from Talk pages, since the Talk page could explicitly discuss the cleanup tags that are supposed to be predicted and would thus lead to biased results.

6 Classification Approach

We regard the problem of quality flaw prediction as a binary classification task. For each flaw, we create a training set that contains a balanced number of positive and negative instances. We use the untagged articles provided in the training corpus as negative instances and select a random subset of the same size as the set of flawed articles.

We use two machine learning algorithms from the Mallet machine learning toolkit [18], a *Naive Bayes* classifier and *C4.5 decision trees*. For efficiently training the Naive Bayes classifier, we perform unsupervised discretization of numeric features using equal interval binning as suggested in [25], since the algorithm does not cope well with real valued features and the Mallet toolkit is not able to perform feature discretization automatically. The decision trees were trained using adaptive boosting with 100 rounds and were limited to the depth of five due to memory restrictions.

7 Evaluation

We experimentally derived the best configuration for each flaw in a *parameter optimization run*, which consists of several training iterations on the same reduced training

[8] `http://snowball.tartarus.org/algorithms/english/stop.txt`

	Advert	Empty Section	Notability	Original Research	Refimprove	Unreferenced	Orphan	Wikify	No Footnotes	Primary Sources	
#Revisions	.008			.020				.008		.006	Revision
#Contributors	.015			.020	.013			.016			
#Registered contributors	.015			.023				.018			
Article age	.007							.012			
Has empty section		**.534**			.004	.007	.002				Structural
Markup to text ratio				.017	.001	.003		.003		.003	
Mean section length		.034				.022	.002	.005	.025		
#Sections	.010	.033			.018	.023	.002		.025	.014	
#References					**.029**	**.250**	.006	.003	**.071**	.004	Reference
#References per sentence					.002	**.250**	.006		**.071**		
References to text ratio				.017		**.250**	.006		**.071**		
Has references											
Has reference list						.013		.003			
#External links	**.067**				.007	.097	.001		.026	**.050**	Network
#Inlinks							.145	.004	.005	.003	
#Outlinks	.013			.002				.011		.007	
Inlinks<3			**.045**	.025			**.472**	**.069**		.002	
No inlinks							.145		.005		
#Organization entities									.015		Named Entity
#Person entities									.006		
#Location entities											
#Organization entities per sentence									.015		
#Person entities per sentence							.002	.003	.006		
#Location entities per sentence							.002	.005		.004	
#Discussions on Talk page	.024			**.144**	.048		.018	.010	.016	.005	Other
#Characters	.021			.031	.005	.003	.003	.008		.012	
#Sentences				.025	.005	.003	.003	.004		.005	
#Tokens	.021			.031			.003	.009		.009	

Table 2: Overview of the feature utility scores per quality flaw. The highest ranked feature for each flaw is written in bold. Lexical features are excluded due to space limitations. Missing values in the tables indicate that the feature has not been selected by the feature selector.

set using different parameters. To this end, we evaluate the performance of both algorithms for each flaw on 10-fold cross validation using 500 positive and 500 negative instances from the training set. We parameterize each run with the number of selected features (between 250 and 1500), the use of a stop-word filter and the frequency cut-off for discarding rare ngrams in order to obtain the best setting.

7.1 Feature Selection

We use the Information Gain feature selection approach [19] to rank and prune the feature space. Table 2 lists all extracted feature types and shows the utility scores which have been determined by the feature selector. The scores depict the discriminativeness of each feature for a given flaw and are the basis for the feature ranking we derived during training. This information sheds light on which types of features work best to represent the individual flaws. It is not surprising that the best indicators for structural flaws are the corresponding structural properties, such as *has empty section* for *Empty Section*. For other flaws, the feature ranking is more interesting. For *Original Research*,

	Advert	Empty Section	Notability	Original Research	Refimprove	Unreferenced	Orphan	Wikify	No Footnotes	Primary Sources	
Classifier	NB	C45	NB	NB	NB	C45	C45	NB	C45	NB	
Selected features	1500	500	250	250	250	1000	1500	1500	250	1500	
false positives	71	73	200	93	71	158	71	250	164	164	Training
false negatives	94	27	63	143	35	51	35	71	74	68	
precision	.850	.863	.679	.792	.590	.741	.858	.635	.715	.714	
recall	.811	.944	.870	.713	.878	.898	.924	.859	.848	.857	
F_1-score	.830	.902	.763	.751	.705	.812	.890	.730	.776	.779	
precision	.853	.876	.661	.740	.615	.780	.863	.678	.730	.736	Test
recall	.826	.912	.852	.767	.751	.884	.925	.844	.902	.866	
F_1-score	.839	.894	.745	.753	.676	.829	.893	.752	.807	.796	

Table 3: Classification performance on training and test set.

for instance, the best ranked feature is the discussion activity. This suggests that the discussion content might also be informative for identifying this flaw and that the Talk pages should be further exploited for feature extraction. For the flaw *Advert*, the most discriminative non-lexical features are links pointing to external resources. Taking into account the context of these external links could further improve the classification performance. It has to be noted that the utility scores cannot be directly compared across flaws. They are only significant as indicators for the ranking within a given flaw. Lexical features are most effective for the flaws *Advert*, *Notability* and *Original Research*, while the other flaws only show little performance gain when adding ngrams to the feature sets. This is to be expected, since structural flaws such as *Empty Section* or *Wikify* are not expressed by the vocabulary but by the article structure and the markup.

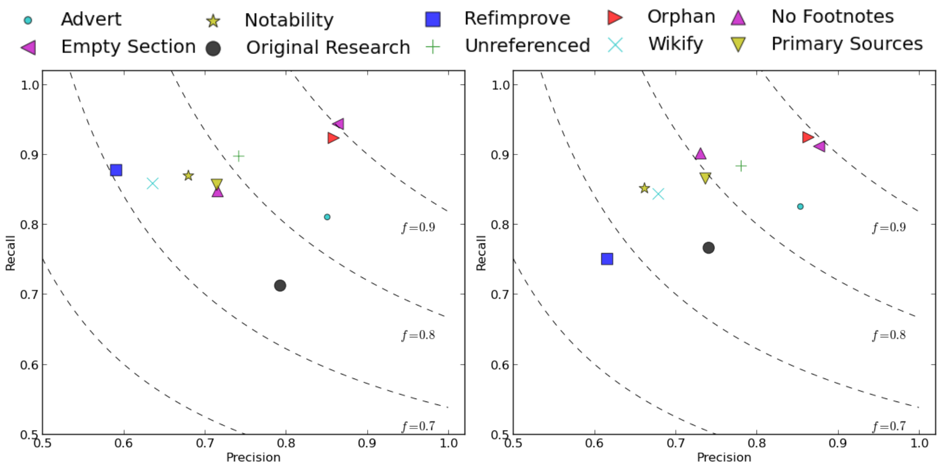
7.2 Classification Performance

Table 3 shows a detailed overview of the classification performance on the training and test set. The results on the training set are based on the best configuration obtained in the parameter optimization run. The performance on the test set has been evaluated by the challenge organizers. The submitted results have been produced using models trained on the whole training set. Furthermore, precision-recall diagrams for both training and test set can be seen in Figure 1.

The good performance on the *Advert* flaw comes surprising, since it initially seemed to be a hard task due to the subjectiveness and subtlety of this flaw. The extracted lexical features are good indicators for the presence of this flaw. On the other hand, the relatively weak performance on *Wikify* was not expected. The prediction of this flaw particularly suffered from the selection of negative instances in the training set. A solution to this is discussed in Section 8.

7.3 Error Analysis

We carried out a detailed error analysis for each flaw in order to identify the main types of errors made by the classifier. The numbers of false positive and false negative instances according to the evaluation on the training set can be seen in Table 3.



(a) (a) Training Set

(b) (b) Test Set

Figure 1: Classifier performance in terms of precision, recall and f-measure

The 71 false positives for *Advert* mostly contain articles about institutions such as universities or government bodies. The descriptions of these institutions resemble the descriptions of companies. However, for companies the same way of writing is more often regarded as advert-style by Wikipedia users than for public institutions. The 94 false negatives are short articles with an average length of 690 tokens. Many of them do not exceed 250 tokens. These articles do not contain enough text to be reliably classified, since the *Advert* flaw largely relies on lexical features.

The 200 false positives for the *Notability* flaw contain a large number of pages about individual persons, organizations, books or movies. Even Wikipedia users have difficulties to judge whether a specific subject qualifies for being included in the encyclopedia. Without world knowledge about the article topic, a reliable judgment cannot be carried out. Furthermore, the notability criteria in Wikipedia are highly disputed in the community and are not interpreted consistently by all users[9]. For a large fraction of the 63 false negatives, the *Notability* template has been removed in newer revisions without a major change of the content (for example in the article on the Bigfoot Trail[10] or the Hong Kong Gold Coast[11]). This suggests that the template has been incorrectly assigned to the training article by the Wikipedia users.

Many of the 158 false positives for the flaw *Unreferenced* did actually have no references at all or just contained an external links section. This suggests that the classifier correctly identified the problem, but the templates were missing in the article. The 51 false negatives are subject to the same problem. In this case, the *Unreferenced*

[9] http://en.wikipedia.org/wiki/Deletionism_and_inclusionism_in_Wikipedia

[10] http://en.wikipedia.org/w/index.php?title=Bigfoot_Trail&diff=502614831&oldid=407680228

[11] http://en.wikipedia.org/w/index.php?title=Hong_Kong_Gold_Coast&diff=502889724&oldid=461337252

template has been used for marking articles that suffer from the *Refimprove* flaw. For example, in the corpus version of the article "Robert Hartmann", the used template was `Unreferenced` but it has been changed to the correct `Refimprove` template in a later version[12]. Similar confusion can be observed in the misclassified instances of the other flaws related to references and citations, such as *Original Research*, *No Footnotes*, and *Primary Sources*. This suggests that the templates should be better defined and consolidated into fewer categories. Other false negative instances for *Unreferenced* are due to the inline usage of the templates. According to the flaw definition, the template applies to articles that do not have any references. However, when used inline in the form `{{Unreferenced|section}}`, it only refers to the section it appears in, while the rest of the article may cite references[13]. In order to account for this, each section has to be classified separately instead of the article as a whole.

According to the instructions provided by the challenge organizers, the *Orphan* template is to be assigned to any article that "has fewer than three incoming links". Therefore, we use the feature `inlinks < 3`, which proved to be the most discriminative one for this flaw. However, the template description in Wikipedia states that articles tagged as *Orphan* have "zero incoming links from other articles"[14]. This discrepancy accounts for most of the false negatives, which have one or two incoming links from other articles. Removing the above mentioned feature and using the inlink counts alone can solve this issue.

The false positives for the flaw *Wikify* mainly consist of short articles. Wikification is not an issue commonly addressed in short articles, and it becomes more important as the article grows. The network and surface features used by the classifier consequently do not work well with short articles.

No regularities could be found for the misclassifications of the flaw *Empty section*. It is likely that the main reason for misclassification are parsing errors. We found that sections containing mainly structured elements such as tables, infoboxes or expanded templates are particularly hard to cope with.

8 Discussion

The work by Anderka et al. [2], in which the authors analyze the evolution of cleanup templates, is a step towards better understanding collaboratively created cleanup tags and how they are utilized by the community. However, the reliability of these labels still remains to be evaluated. Future work has to address the issue of annotation quality in a controlled annotation study reporting the agreement of the annotators with the community created gold standard. The results of this study will reveal the upper bound for an automatic classification task. Only then will it be possible to draw sound conclusions from analyses based on this data and to put classification results into perspective.

[12] http://en.wikipedia.org/w/index.php?title=Robert_Hartmann&diff=474150162&oldid=466987161

[13] for example in http://en.wikipedia.org/w/index.php?title=White_Oleander_%28film%29&oldid=463206537

[14] http://en.wikipedia.org/wiki/Wikipedia:Orphan#Criteria

In this work, we chose a binary classification approach for predicting quality flaw labels. While binary classification is supported by many mature algorithms for which high-performance implementations are readily at hand, it poses the problem of selecting discriminative negative instances. This issue has been critically discussed by Anderka et al. [3]. Even though we found the classification performance to be good when using a random sample of untagged articles as negative instances, a more sophisticated selection technique might improve the results even more. Since no articles are available that are explicitly tagged as not suffering from a certain flaw, the *removal* of a flaw marker might indicate that a specific article revision does no longer suffer from the same flaw. Consequently, article revisions that have just lost a specific flaw marker might serve as more discriminative negative instances for training a binary classifier for the flaw.

Acknowledgments

We thank Artem Vovk and Michael Kutschke for implementing and testing parts of the software. This work has been supported by the Volkswagen Foundation as part of the Lichtenberg-Professorship Program under grant No. I/82806, and by the Hessian research excellence program "Landes-Offensive zur Entwicklung Wissenschaftlich-ökonomischer Exzellenz" (LOEWE) as part of the research center "Digital Humanities"

References

1. Anderka, M., Stein, B.: A Breakdown of Quality Flaws in Wikipedia. In: 2nd Joint WICOW/AIRWeb Workshop on Web Quality. pp. 11–18. Lyon, France (2012)
2. Anderka, M., Stein, B., Busse, M.: On the Evolution of Quality Flaws and the Effectiveness of Cleanup Tags in the English Wikipedia. In: Wikipedia Academy 2012. Berlin, Germany (2012)
3. Anderka, M., Stein, B., Lipka, N.: Predicting Quality Flaws in User-generated Content: The Case of Wikipedia. In: 35th International ACM Conference on Research and Development in Information Retrieval (SIGIR 12) (2012)
4. de Castilho, R.E., Gurevych, I.: A Lightweight Framework for Reproducible Parameter Sweeping in Information Retrieval. In: Proceedings of the Workshop on Data Infrastructures for Supporting Information Retrieval Evaluation. pp. 7–10. Glasgow, UK (2011)
5. Dalip, D.H., Gonçalves, M.A., Cristo, M., Calado, P.: Automatic Quality Assessment of Content Created Collaboratively by Web Communities. In: Proceedings of the Joint International Conference on Digital Libraries. pp. 295–304. Austin, TX, USA (Jun 2009)
6. Dohrn, H., Riehle, D.: Design and implementation of the Sweble Wikitext parser. In: Proceedings of the 7th International Symposium on Wikis and Open Collaboration. pp. 72–81. Mountain View, CA, USA (2011)
7. Dohrn, H., Riehle, D.: Wom: An object model for wikitext. Tech. rep., University of Erlangen, Erlangen, Germany (2011)
8. Ferrucci, D., Lally, A.: UIMA: An Architectural Approach to Unstructured Information Processing in the Corporate Research Environment. Natural Language Engineering 10(3-4), 327–348 (2004)
9. Ferschke, O., Gurevych, I., Chebotar, Y.: Behind the Article: Recognizing Dialog Acts in Wikipedia Talk Pages. In: Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics. pp. 777–786. Avignon, France (Apr 2012)

10. Ferschke, O., Zesch, T., Gurevych, I.: Wikipedia Revision Toolkit: Efficiently Accessing Wikipedia's Edit History. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. System Demonstrations. pp. 97–102. Portland, OR, USA (Jun 2011)
11. Finkel, J.R., Grenager, T., Manning, C.: Incorporating non-local information into information extraction systems by Gibbs sampling. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. pp. 363–370. Association for Computational Linguistics, Morristown, NJ, USA (Jun 2005)
12. Giles, J.: Internet encyclopaedias go head to head. Nature 438(7070), 900 (2005)
13. Han, J., Fu, X., Chen, K., Wang, C.: Web Article Quality Assessment in Multi-dimensional Space. In: Proceedings of the 12th International Conference on Web-age Information Management, pp. 214–225. Lecture Notes in Computer Science, Wuhan, China (2011)
14. Han, J., Wang, C., Jiang, D.: Probabilistic Quality Assessment Based on Articles Revision History. In: Proceedings of the 22nd International Conference on Database and Expert Systems Applications. pp. 574–588. Toulouse, France (2011)
15. Hu, M., Lim, E., Sun, A., Lauw, H., Vuong, B.: Measuring article quality in wikipedia: models and evaluation. In: Proceedings of the 16th ACM Conference on Information and Knowledge Management. pp. 243–252. CIKM '07, Lisbon, Portugal (2007)
16. Javanmardi, S., Lopes, C.: Statistical measure of quality in Wikipedia. In: Proceedings of the First Workshop on Social Media Analytics - SOMA '10. pp. 132–138. Washington DC, DC, USA (2010)
17. Lipka, N., Stein, B.: Identifying featured articles in wikipedia. In: Proceedings of the 19th International Conference on World Wide Web. p. 1147. Raleigh, NC, USA (Apr 2010)
18. McCallum, A.K.: MALLET: A Machine Learning for Language Toolkit (2002), `http://mallet.cs.umass.edu`
19. Mitchell, T.: Machine Learning. McGraw-Hill Education (ISE Editions), 1st edn. (1997)
20. Ogren, P.V., Wetzler, P.G., Bethard, S.: ClearTK: A UIMA toolkit for statistical natural language processing. In: Towards Enhanced Interoperability for Large HLT Systems: UIMA for NLP workshop at Language Resources and Evaluation Conference (LREC) (2008)
21. Rassbach, L., Pincock, T., Mingus, B.: Exploring the Feasibility of Automatically Rating Online Article Quality. Proceedings of the 9th Joint Conference on Digital Libraries (2007)
22. Stvilia, B., Gasser, L., Twidale, M.B., Smith, L.C.: A Framework for Information Quality Assessment. Journal of the American Society for Information Science 58(12), 1720–1733 (2007)
23. Stvilia, B., Twidale, M.B., Smith, L.C., Gasser, L.: Information Quality Work Organization in Wikipedia. Journal of the American Society for Information Science and Technology 59(6), 983–1001 (Apr 2008)
24. Wilkinson, D.M., Huberman, B.A.: Cooperation and Quality in Wikipedia. In: Proceedings of the 2007 International Symposium on Wikis. pp. 157–164. Montreal, Canada (2007)
25. Witten, I.H., Frank, E., Hall, M.A.: Data mining : Practical Machine Learning Tools and Techniques. Morgan Kaufmann, Burlington, MA (2011)
26. Wu, K., Zhu, Q., Zhao, Y., Zheng, H.: Mining the Factors Affecting the Quality of Wikipedia Articles. In: Proceedings of the 2010 International Conference of Information Science and Management Engineering. pp. 343–346 (Aug 2010)
27. Yaari, E., Baruchson-Arbib, S., Bar-Ilan, J.: Information quality assessment of community generated content: A user study of Wikipedia. Journal of Information Science 37(5), 487–498 (Aug 2011)
28. Zesch, T., Müller, C., Gurevych, I.: Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary. In: Proceedings of the 6th International Conference on Language Resources and Evaluation. Marrakech, Morocco (2008)