

Assessing the quality of Wikipedia articles

Abstract

For our project we experimented with using binary classifiers to distinguish good articles from bad ones at Wikipedia. We describe the current peer review processes at Wikipedia, our methods for collecting data, the features we chose to examine, and our experiments with various classifiers. The classifiers were trained using the quality judgments made by peer reviewers, and were tested (using cross-validation) against these same judgments. We found that we were able to achieve quite low error rates using a Random Forests classifier.

Issues of quality on Wikipedia

In the past few years, the Wikipedia project has grown from an experiment in open, collaborative editing to one of the most trafficked sites on the web. As bloggers, news sites and even graduate student instructors have begun adopting it as the site to use when linking readers to more information on a subject, it has come to dominate search rankings: a recent study showed that it appeared in the top ten search results for queries on topics it covers 65% of the time.¹ But critics have raised questions about the quality of the articles on Wikipedia. In 2005, after a high-profile incident involving a false biography on Wikipedia,²

¹ Jure Cuhalev, "Ranking of Wikipedia articles on search engines for searches about its own articles," <http://www.kiberpipa.org/~gandalf/blog-files/wikistatus/wikistatus.pdf>.

² http://en.wikipedia.org/wiki/John_Seigenthaler_Sr._Wikipedia_biography_controversy.

many within the Wikipedia community also began taking a closer look at their processes for ensuring quality. In 2006 Larry Sanger, one of the founders of the Wikipedia, announced his plans to fork Wikipedia and create a new project called Citizendium, which would require peer review by experts before articles were made publicly viewable.³

The Wikipedia peer review processes

Though they do not necessarily involve certified experts, Wikipedia has its own peer review processes. One set of processes seeks to identify articles of high quality, so that they can be spotlighted or selected for inclusion in curated snapshots.⁴ The cream of the crop are named “featured articles,” some of which are shown on the front page of Wikipedia for a day. Articles that have been nominated to be featured are judged on the basis of clarity, comprehensiveness, accuracy, and neutrality, and they must have achieved stability in terms of revisions made.⁵ The review process is rather involved, and reviewers must reach consensus (not just a simple majority) before featured article status can be granted. As of December 2006, there are approximately 1200 featured articles among the more than 1.5 million total articles at Wikipedia—under 0.1%.

A less stringent process for recognizing quality allows anyone to recommend an article for “good” status, which can then be granted by another (impartial) reviewer. The criteria for good articles are less exacting than those for featured articles, focusing more on the structural and organizational best practices established by the prevailing Wikipedia guidelines. Despite this, there are not many more good articles than there are featured articles, around 1500 out of 1.5 million (about 0.1%). It seems that even the streamlined

³ <http://citizendium.org/>.

⁴ A simplified version of Wikipedia's review process is presented here. In actuality, there are any number of sub-projects at Wikipedia devoted to reviewing articles in specific subject domains like “Australia,” and each of these projects may have its own criteria and processes for judging articles.

⁵ http://en.wikipedia.org/wiki/Wikipedia:What_is_a_featured_article.

review process does not have enough participation to process more than a small fraction of Wikipedia content.

Another set of processes aims to weed out articles that have been deemed inappropriate for inclusion in Wikipedia, whether for reasons of quality or subject matter. Just as there both full and streamlined versions of the quality recognition processes, so are there “regular” and “speedy” deletion processes. The latter is meant for recurrent spam problems and the like, while the former is very similar to the featured article process: nomination for deletion followed by a period for consensus decision-making.⁶ Alternatives to deletion include being merged with related articles or moved to a different Wikimedia project such as Wiktionary.

While all these processes seem to be more or less functional, they are rather slow and prone to backlogs. It is not clear that they will scale with the growth of Wikipedia. For this project, we investigated whether a classifier might be trained to recognize the features of a high-quality or low-quality Wikipedia article, and thus provide an “first pass” filter for peer review processes. Such a classifier might be used to build a web service which would return classification predictions for specified article IDs. Browser scripts could then query this service to indicate the probably quality of articles being viewed. Wikipedians already have an arsenal of similar tools to help them with maintenance tasks;⁷ a successful article quality classifier would be a useful addition to these.

Data collection

Our data was collected via the MediaWiki API,⁸ a REST-style web service maintained by Wikipedians, which can be queried for article content and metadata in XML

⁶ http://en.wikipedia.org/wiki/Wikipedia:Articles_for_deletion.

⁷ http://en.wikipedia.org/wiki/Wikipedia:WikiProject_User_scripts.

⁸ <http://en.wikipedia.org/w/api.php>.

format. This spared us the pain of having to scrape Wikipedia HTML pages. Unfortunately, the API is still under development, so some of our queries were more difficult than expected. For example, currently the API does not support querying for articles by category. This meant that we could not easily obtain a list of articles marked for deletion, and instead had to parse the content of a Wikipedia page which contains this list, and query for the article metadata one at a time.

There are also turned out to be no simple way to get a list of “featured” or “good” articles via the API. Though there are pages listing these articles, parsing their content is tricky and error prone due to the complex MediaWiki markup. To obtain our lists, we first queried the API for articles which embed the “good” and “featured” badges. Because anyone might illicitly add one of these badges to their article, we then checked each title in the list by searching for it in the official page listing “good” or “featured” articles. After obtaining lists of featured, good, and marked for deletion articles in this way, we queried the API for each article's full revision history (including editor and timestamp of each revision), as well as the full list of other Wikipedia pages linking to that article (backlinks). Querying for this data was rather slow, as the Wikipedia API only returns results 50 at a time and has considerable latency. It took a few days before we had a script stable enough to run overnight gathering data without crashing due to network problems or invalid XML. The XML was parsed and stored as rows in a MySQL database to ease further feature calculation.

Data overview

We ended up with 1136 featured articles, 1582 good articles, and 378 articles marked for deletion, with 2.5 million revisions and 2.1 million backlinks among them. As

expected, the articles exhibited very skewed distributions for a number of the features we were interested in. As the graphs of the distributions for backlinks and revisions show, very few articles attract an extraordinary amount of activity, while most have very little.

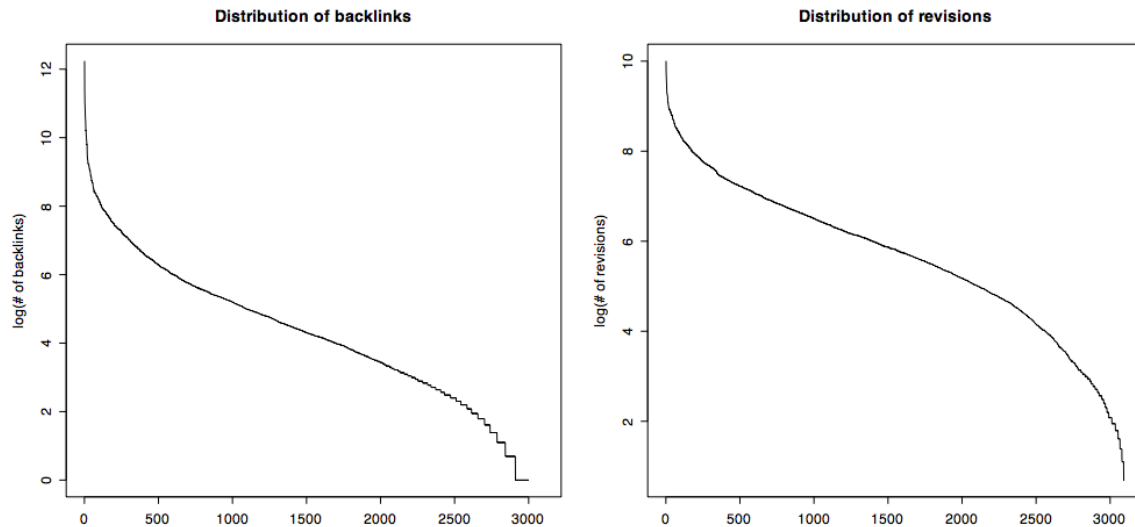


Figure 1.

Yet there are also some fairly clear differences among the classes. Articles marked for deletion have very few backlinks or revisions on average, while featured articles and good articles have many more. Furthermore, there is a small but significant difference in the distributions of these features between good article and featured articles. As we show in the following sections, these differences bode well for our classifier experiments.

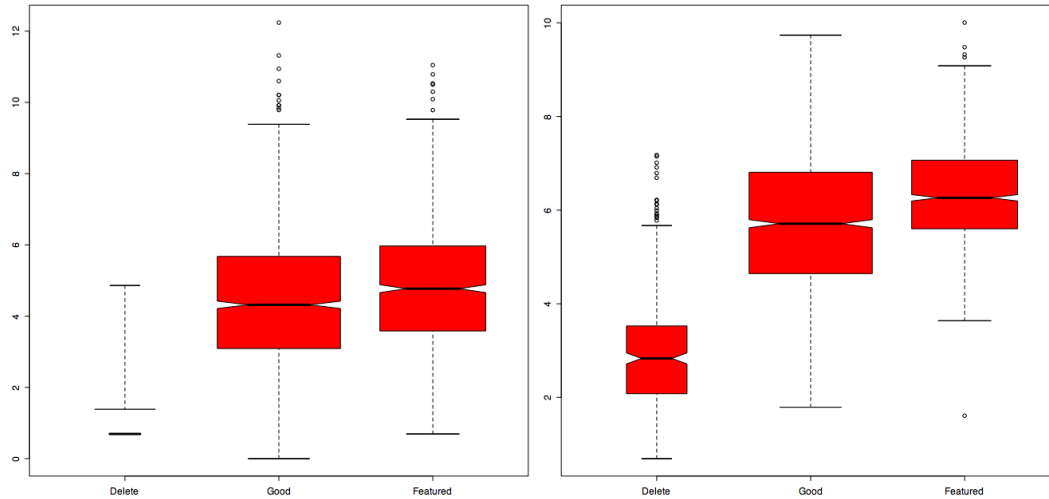


Figure 2.

Feature Generation

With a substantial dataset of articles and revisions, we moved on to extracting additional features. We used the Python programming language to calculate the 21 features from our dataset. The goal of our classifier was to identify good and bad articles within the context of Wikipedia. We believe that structural features (e.g., number of headings) and implicit features (e.g., number of editors and back links) are sufficient to characterize Wikipedia articles. The underlying assumption is that editors will work to make worthwhile articles conform to Wikipedia standards for good articles by adding images, internal links, defining sections, et cetera. The following table describes each of the features we used for classification.

Feature	Description
page_byte_size	The size of the current article revision (including all markup) in bytes. While no one size is appropriate for all articles, Wikipedia guidelines recommend articles should not be larger than 32KB. Very small articles, on the order of 3KB or smaller, were more likely to be bad.
revision_count	The number of revisions made to the article. If we assume that each revision improves the article, then more revisions should favor better articles.
revision_last_month	The number of revisions in the last 30 days. Wikipedia guidelines for featured say the article should be stable. Stability in this context means there is a low volume of revisions and the revisions represent minor changes.
editor_count	The number of editors who have made revisions to the article. Bad articles tended to have very few editors compared to good and featured articles.
backlink_count	The number of articles in Wikipedia that link to the article. Articles in the "featured" class had significantly more backlinks on average. This feature favors articles that can be referenced broadly, across domains. For example, articles about locations tend to have higher than average backlink counts. It is also possible that featured articles are better known to Wikipedia contributors and therefore more likely to be linked in other articles.
outbound_internal_links	The number of links to other articles within Wikipedia. Wikipedia relies heavily on interlinked articles to help users navigate information within a given topic area. Good and featured articles had more internal links on average. As one would suspect, we noted a correlation between article size and outbound internal links.
outbound_external_links	The number of links to webpages outside of Wikipedia. Outbound external links are used extensively as references on Wikipedia.
image_count	The number of images in the article. Most good and featured articles had at least one image.
heading_count	The number of headings within an article. Headings are indicated with wiki markup and used on every good and featured page. It is common to see multiple levels of headings, however, this feature ignores heading level when counting.
special_tag_count	The number of special tags. Special tags are the means by which users mark pages for things like deletion or featured status. Special tags are also used to add preformatted content to a page.
has_references	1 if the article has a "References", "Notes", or "Bibliography" heading, otherwise 0. Most good and featured articles have one or more of these sections. It is generally accepted that all good Wikipedia articles need a section for citations.
has_external_links	1 if the article has an "External Links" heading, otherwise 0. About 66% of good and featured articles have this section and the opposite is true for bad articles.
has_see_also	1 if the article has a "See Also" heading, otherwise 0. About half of the good and featured articles have this section, but only about 10% of the bad articles have it.
fog	Gunning Fog Index. A readability test designed to gauge the understandability of a text. Its output is an approximate representation of the U.S. grade level needed to comprehend the text. Wiki markup was stripped from the text before running the algorithm.
ari	Automated Readability Index. A readability test designed to gauge

Feature Selection

In our classification results, we used all the features listed above. However, we ran various feature selection algorithms to determine which features were most useful. The following graph shows the output of Information Gain Ranking Filter in WEKA. Backlink count comes ahead of the other features, suggesting that the volume of links from other Wikipedia pages is a strong indicator. All seven of the readability metrics proved to be largely inconsequential, comparatively. In future work, we would not bother using more than two readability metrics.

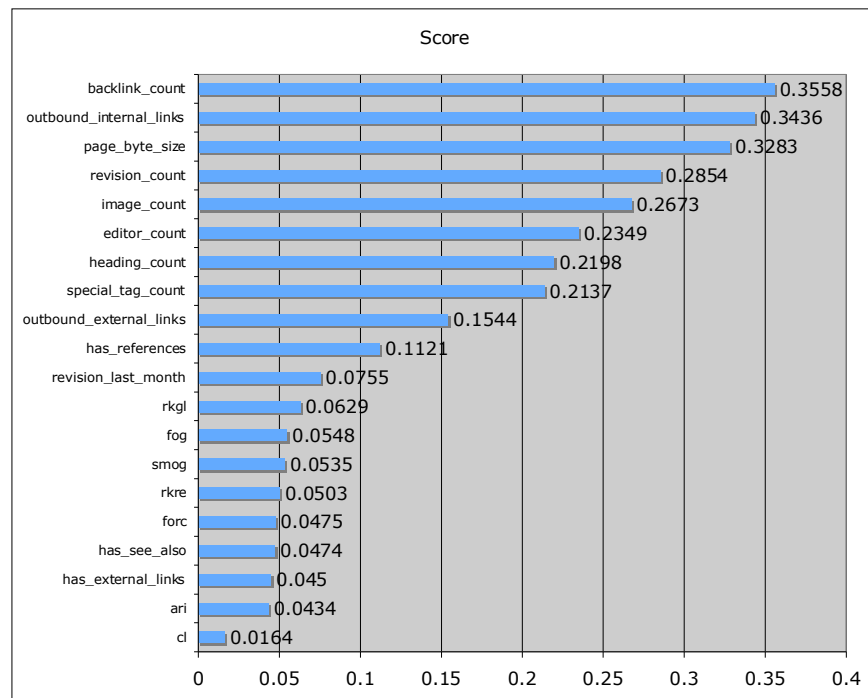


Figure 3.

WEKA's CFS attribute subset evaluator with exhaustive search gave us similar results. The recommended set of features included: `revision_count`, `backlink_count`,

page_byte_size, outbound_internal_links, image_count, special_tag_count, has_references, has_see_also, fkgf.

Results

We used the WEKA Explorer environment for performing learning over our feature set. We chose to perform a binary classification (good/featured vs. bad articles). As such, we have removed the article_class_threeway attribute from experimentation. We considered four learning algorithms: two simple/popular ones (k-Nearest Neighbors, Naïve Bayes) and two more advanced ones (Random Forest, SVM). The results are summarized in the following sections.

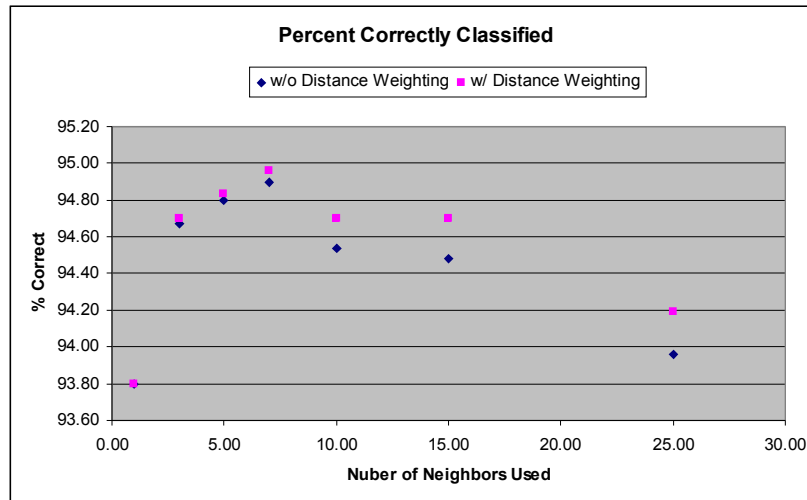
1. *k-Nearest Neighbors (IBk)*

This is among the simplest learning algorithms. We simply assign classification based on the class assignments of the k-nearest data points. If $k=1$, then we assign based on the single nearest neighbor. We can also weigh the contribution of each neighbor by its distance ($\text{weight}=1/\text{distance}$). The results are summarized below.

Number of Neighbors	Correctly Classified Percentage (without / with distance weighting)
1	93.80 / 93.80
3	94.67 / 94.70
5	94.80 / 94.83
7	94.90 / 94.96
10	94.54 / 94.70
15	94.48 / 94.70
25	93.96 / 94.19

Overall, kNN performs extremely well. We observe that using distance weighing results in slightly higher percentage of correctly classified points. We also observe

the optimal number of neighbors lies somewhere about 7 (see below).



2. Naïve Bayes (Updatable Classifier)

This is another simple learning algorithm, but based on estimator classes. We tried the algorithm both with and without using the kernel estimator. The performance was not as good as kNN, but still respectable.

Use Kernel Estimator	Correctly Classified Percentage
No	73.74
Yes	83.46

3. Random Forest

The random forest learning algorithm is a classifier consisting of many decision trees. We tried the algorithm for varying number of trees. The performance was excellent, and seemed independent of the number of trees used.

Number of Trees	Correctly Classified Percentage
5	98.19
10	98.48
15	98.42
20	98.48

4. Support Vector Classifier (SMO)

For the support vector classifier, we have chosen to vary the complexity parameter (C) and whether RBF kernel would be used instead of a polynomial one. We observe that this learning algorithm performed slightly better than kNN, but not quite as good as the random forests above. In addition, the polynomial kernel appears to perform slightly better than the RBF kernel.

Complexity (C)	Correctly Classified Percentage (without RBF / with RBF)
0.05	87.79 / 87.79
0.5	94.57 / 87.79
1.0	94.80 / 87.79
1.5	94.99 / 87.79
5	95.22 / 91.86
50	95.28 / 94.77

Analysis of Random Forest Results:

In this section, we will examine the results of using the Random Forest algorithm with 10 trees in more detail. The results from WEKA are:

=== Summary ===

Correctly Classified Instances	3049	98.4819 %
Incorrectly Classified Instances	47	1.5181 %
Kappa statistic	0.9269	
Mean absolute error	0.0246	
Root mean squared error	0.1107	
Relative absolute error	11.4702 %	
Root relative squared error	33.8125 %	
Total Number of Instances	3096	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.996	0.098	0.987	0.996	0.991	good
0.902	0.004	0.972	0.902	0.936	bad

=== Confusion Matrix ===

```
a  b  <-- classified as
2708 10 | a = good
37 341 | b = bad
```

We observe from the confusion matrix that a small number of good articles were classified as bad and vice versa. This FP is about 1% for good articles and 0.4% for bad articles.

Though, we should also note that there are far more good articles in our dataset than there are bad articles. Recall that bad articles are those that are marked for deletion, and in fact, the majority of Wikipedia articles are obviously not marked for deletion.

It is useful to consider the features identified previously as high information gain such as `backlink_count`, `outbound_internal_links`, `page_byte_size`, and `revision_count`. For example, consider the attribute `page_byte_size` (see figure below). Each article is plotted as a single point whose page size is both the x and y coordinates. Hence, all the articles will lie on the diagonal. Next the good articles are colored blue and the bad articles are colored red. As can be seen from the figure, bad articles almost always have smaller byte sizes (reside in the lower left hand corner) than good articles. We find similar pictures for `backlink_count`, `outbound_internal_links`, and `revision_count`.

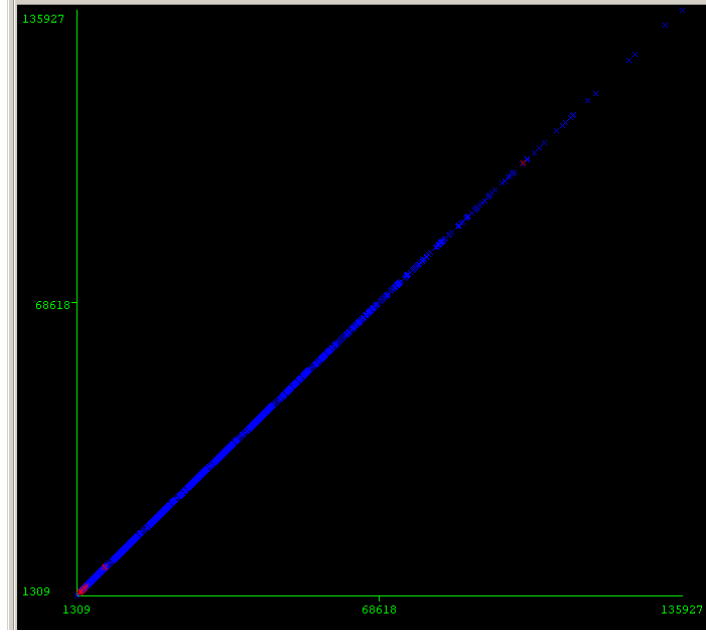


Figure 4: Page byte size (good- blue, bad- red)

Future Work

Our current feature set proved effective, however, we believe much work could be done to improve it. First, the use of raw counts may benefit from normalization against article length. Second, we have only superficially attempted to incorporate the temporal component of revision history. We suspect that analyzing revision history over time would yield valuable features. Finally, examining attributes of article contributors that could represent “experience,” such number of edits, may help, in particular with differentiating articles that are well constructed, but not appropriate for Wikipedia.

Aside from better features, we believe a more robust dataset, one including more bad articles and mediocre articles, would help us to fine tune our classifier. The good and featured class and the bad class are very different from each other, so it is difficult to know how well our classifier works around edge cases.