

Wiki-Reliability: A Large Scale Dataset for Content Reliability on Wikipedia

KayYen Wong*
Outreachy
Kuala Lumpur, Malaysia
kayyenwong@gmail.com

Miriam Redi
Wikimedia Foundation
London, United Kingdom
miriam@wikimedia.org

Diego Saez-Trumper
Wikimedia Foundation
Barcelona, Spain
diego@wikimedia.org

ABSTRACT

Wikipedia is the largest online encyclopedia, used by algorithms and web users as a central hub of reliable information on the web. The quality and reliability of Wikipedia content is maintained by a community of volunteer editors. Machine learning and information retrieval algorithms could help scale up editors' manual efforts around Wikipedia content reliability. However, there is a lack of large-scale data to support the development of such research. To fill this gap, in this paper, we propose *Wiki-Reliability*, the first dataset of English Wikipedia articles annotated with a wide set of content reliability issues.

To build this dataset, we rely on Wikipedia "templates". Templates are tags used by expert Wikipedia editors to indicate content issues, such as the presence of "non-neutral point of view" or "contradictory articles", and serve as a strong signal for detecting reliability issues in a revision. We select the 10 most popular reliability-related templates on Wikipedia, and propose an effective method to label almost 1M samples of Wikipedia article revisions as *positive* or *negative* with respect to each template. Each positive/negative example in the dataset comes with the full article text and 20 features from the revision's metadata. We provide an overview of the possible downstream tasks enabled by such data, and show that *Wiki-Reliability* can be used to train large-scale models for content reliability prediction. We release all data and code for public use.

CCS CONCEPTS

• **Information systems** → **World Wide Web; Information retrieval**; • **Computing methodologies** → **Machine learning**.

KEYWORDS

Wikipedia, Dataset, NLP, Reliability, Hoaxes, POV, Unreferenced

ACM Reference Format:

KayYen Wong, Miriam Redi, and Diego Saez-Trumper. 2021. Wiki-Reliability: A Large Scale Dataset for Content Reliability on Wikipedia. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, June 2021. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3404835.3463253>

*This work was done during KayYen's internship at the Wikimedia Foundation.



This work is licensed under a Creative Commons Attribution International 4.0 License.

SIGIR '21, July 11–15, 2021, Virtual Event, Canada

© 2021 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-8037-9/21/07.

<https://doi.org/10.1145/3404835.3463253>



Figure 1: Example of an English Wikipedia page with several template messages describing reliability issues.

1 INTRODUCTION

Wikipedia is one the largest and most widely used knowledge repositories in the world. People use Wikipedia for studying, fact checking and a wide set of different information needs [11]. In developing countries, Wikipedia is largely used as an educational resource [8]. Moreover, not only people, but AI-based systems learn from Wikipedia [9] and use it as ground-truth for fact checking [12]. Therefore, the quality of content on Wikipedia is relevant to both humans and machines, and more broadly for the integrity of all human knowledge.

On Wikipedia, the review and moderation of content is self-governed by Wikipedia's volunteer community of editors, through collaboratively created policies and guidelines [2, 5]. Despite the large size of the Wikipedia editors community (41k monthly active editors for English Wikipedia in 2020¹), the labor cost of monitoring Wikipedia content quality and patrolling new edits (around 200k daily on English Wikipedia) is intensive. Automated strategies could be used to support the community of Wikipedia editors and reduce their workload, allowing editors to focus their efforts on more complex content moderation efforts. Despite recent advancements in the fields of Information Retrieval (IR) and Natural Language Processing (NLP), there exists only a few examples of successful automated support systems for Wikipedia content reliability monitoring, the most well-known being ORES [6], an open source service using multiple independent Machine Learning classifiers to score Wikipedia edits in real time.

One of the main reasons for this gap is the lack of training data that researchers can use to detect and resolve content quality matters. To encourage research advancements on this front, in

¹[https://stats.wikimedia.org/#/en.wikipedia.org/contributing/active-editors/normal|line|2020-01-01~2021-01-01|\(page_type\)-content*non-content|monthly](https://stats.wikimedia.org/#/en.wikipedia.org/contributing/active-editors/normal|line|2020-01-01~2021-01-01|(page_type)-content*non-content|monthly)

this paper we propose *Wiki-Reliability*, a large dataset of English Wikipedia articles annotated with a wide range of content reliability issues.

To create this dataset, we rely on Wikipedia’s *maintenance templates*, one of the key mechanisms for Wikipedia editors to monitor content quality. Templates appear as messages on articles, warning for quality issues within a page content (see Figure 1). The usage of templates requires expert knowledge of Wikipedia community processes. Therefore, templates can be considered as expert-created labels, and the implicit data created as a by-product of Wikipedia editors workflow as reasonably high-quality crowd generated dataset.

Wiki-Reliability focuses on the list of templates curated by the WikiProjectReliability² editors, who maintain templates related to citations and verifiability issues, which are used to signal that moderation fixes are needed to improve article reliability.

To create *Wiki-Reliability*, we propose a novel methodology for template-based article labeling. This method can be easily reproduced to create new datasets using other types of templates on Wikipedia. For each reliability template, we extract pairs of *positive* and *negative* versions of a Wikipedia article from its revision history. Positive examples are versions of an article which contain a reliability issue, signalled by the *addition* of the template, while negative examples are article revisions where the issue has been resolved, signalled by the *removal* of the template.

Together with the positive/negative labels, for each sample in our dataset we also provide a set of metadata features, which help contextualize each data point, adding information that is not directly related with the textual content, for example the number of external links in an article, or the number of links pointing to other Wikipedia pages. We also parse the full article textual content, and include it as part of the dataset. We then identify a set of downstream research tasks enabled by the different template labels in our dataset. Finally, we test baselines for content reliability prediction, showing that the features in our data are informative, but that downstream tasks are not trivial to solve and require further research efforts.

NLP and IR practitioners and researchers use Wikipedia extensively. This data will provide the opportunity for these communities to contribute back to Wikipedia, and help improve the quality of the online encyclopedia, while experimenting with large-scale data analysis techniques that can be generalized to other contexts across the World Wild Web. All the data and code generated for this project is being released with this paper.³

2 METHODOLOGY FOR DATA LABELING

In this Section, we explain our proposed methodology to effectively extract high-quality annotated data from the unstructured content of Wikipedia articles.

2.1 Background: Wikipedia, Revisions, Templates and Wikiprojects

Content on Wikipedia is dynamic. After an article is created, editors contribute by generating updated versions. These versions are known as *revisions*. Each revision has a numerical id, and is

stored in the editing history of the article. Revisions also come with associated metadata, such the timestamp, revisions’ author and a comment added by the author summarizing the aim of their contribution.

Wikipedia *Templates*, also known as transclusions, are defined as “pages created to be included in other pages”⁴. Editors use templates for a wide set of operations, such as creating infoboxes, navigational boxes, or creating warning messages. One of the most well known templates is the “citation needed” tag used to signal content requiring a citation.

Wikiproject is the concept used by Wikipedia editors to name groups of editors that collaborate on specific tasks. There are topic-focused Wikiprojects such *WikiProject Science* or *Wikiproject Sports*. But there are also groups of users that congregate to add and improve images and multimedia content to articles across topics, or to disambiguate page titles. In this paper we focus one group of users working on detecting and fixing problems related to content reliability.

2.2 Selection of Templates

To annotate Wikipedia article revisions, we start from the list of 41 templates curated by the WikiProjectReliability. We categorize them according to their *span*, namely the coverage of the template, i.e., if it indicates a reliability issue at *article*, *sentence* or *section* level. We then manually sort the article-level maintenance templates based on their impact to Wikimedia, prioritizing templates which are of interest to the community.

2.3 Parsing Wikipedia dumps

Next, we search for the presence of templates in Wikipedia revisions. The full history of Wikipedia articles is available through periodically updated XML dumps⁵. We use the full English Wikipedia dump from September 2020, sizing 1.1TB compressed in bz2 format. We convert this data to AVRO⁶ and process it using PySpark. We apply a regular expression to extract all the templates in each revision, and retain all the articles that contain our predefined list of templates. Next, using the MediaWiki History dataset[4] we obtain additional information (metadata) about each revision of the aforementioned articles.

2.4 Handling instances of vandalism

The accurate detection of true positive and negative cases is further complicated by instances of vandalism, namely where a template has been maliciously/wrongly added or removed. To handle this, we rely on the wisdom of the crowd by ignoring revisions which have been reverted by other editors.

Research by [7] suggests that 94% of reverts can be detected by matching MD5 checksums of revision content historically. However, comparing SHA checksums of consecutive revisions is a computationally expensive process as it requires processing through the entire history of revisions. Fortunately, the *MediaWiki History*⁷ table contains monthly data dumps of all events with pre-computed

²https://en.wikipedia.org/wiki/Wikipedia:WikiProject_Reliability

³<https://github.com/kay-wong/Wiki-Reliability/>

⁴<https://en.wikipedia.org/wiki/Help:Template>

⁵<https://dumps.wikimedia.org>

⁶<https://github.com/wikimedia/analytics-wikihadoop>

⁷https://wikitech.wikimedia.org/wiki/Analytics/Data_Lake/Edits/MediaWiki_history

Template	Count	Span	Description	Downstream Tasks
<i>Unreferenced</i>	389966	article	Article has no citations or references at all	SR
<i>One Source</i>	25085	article	Article cites only a single source.	SR
<i>Original Research</i>	19360	article	Article contains original research.	CRP
<i>More Citations Needed</i>	13707	article	Article needs additional citations for verification.	CRP, SR
<i>Unreliable Sources</i>	7147	article	Some of the article’s listed sources may not be reliable.	CRP, SR
<i>Disputed</i>	6946	article	Article’s factual accuracy is disputed.	CRP
<i>Pov</i>	5214	article/section	Article lacks a neutral point of view.	CRP
<i>Third-party</i>	4952	article	Article relies excessively on sources too closely associated with the subject	CRP, SR
<i>Contradict</i>	2268	article/section	Article or section contradicts itself	CRP
<i>Hoax</i>	1398	article	The truthfulness of an article is questioned, and is believed to constitute a hoax.	CRP

Table 1: List of revision templates in the Wiki-Reliability dataset, together with the positive/negative pair counts, description, and ideas for potential downstream tasks (CRP= Content Reliability Prediction, SR= Source Retrieval).

fields of computationally expensive features to facilitate analyses. We use the *revision_is_identity_reverted* feature in revisions events, which marks whether a revision was reverted by another future revision, and exclude all reverted revisions from the articles’ edit history.

2.5 Obtaining positive and negative examples

We next iterate through all consecutive revisions of articles’ edit history to extract positive/negative class pairs for each template.

We repeat the following process for all templates in our list. We retain as *positive* the first revision where a content-reliability template appears. Next, we look for *negative* examples. Labeling negative samples in this context is a non-trivial task. While the presence of a template is a strong signal of a content reliability issue, the absence of a template does not necessarily imply the converse. A revision may contain a reliability issue that has not yet been reviewed by expert editors and flagged with a maintenance template. Therefore, we iterate through the article history succeeding the positive revision, and label as *negative* the first revision where the template does not appear, i.e. the revision where the template was *removed*. The negative example then acts as a contrasting class, as it constitutes the positive example which has been fixed for its reliability issue.

This approach is particularly effective to ensure data coverage and quality. While previous work selects positive and negative examples as pages *with* and *without* templates from a current snapshot of pages[1, 3], in our approach, we capture multiple instances of templates’ additions and removals over the full history of Wikipedia articles. By extracting pairs of positive/negative examples from the same article, we eliminate confounding factors such as the article topic, or its inherent quality, and provide data which is balanced across classes.

2.6 Summary of the Data Labeling Pipeline

- (1) Obtain Wikipedia edit history: We download the Wikipedia dump available at September 2020⁸, and load it in AVRO format for PySpark processing.
- (2) Obtain “reverted” status of a revision: We query the Mediawiki History dataset to extract the “reverted” status of a revision.

⁸<https://dumps.wikipedia.org>

- (3) Check if the revision contains a template: For each template in the template list, we loop over all non-reverted revisions to find the first revision where the template has been added. We mark such revision as *positive* to indicate that the revision contains the template.

- (4) Process positive/negative pairs:

We iterate through all consecutive revisions of a positive example to find the next non-reverted revision where the template has been removed, and mark it as *negative*.

3 DATASET DESCRIPTION

We explain here the structure and post-processing of the final dataset released. After replicating the process in Section 2.6 for all the article-level templates, we select the templates with the largest amount of annotations (Section 3.1), extract the corresponding revision text, and compute metadata features (Sections 3.4 and 3.3, respectively). The final dataset includes, for each template, the following information: page identifier, revision identifier, positive/negative template labels, text and features, as shown in Table 2. We compiled two files for each template, one with annotations and metadata, and another one with the full revision text. The data is publicly available on Figshare⁹.

3.1 Reliability Templates and Labels

We repeat the positive/negative revision labeling for all the templates curated by WikiProjectReliability, and assign, to each matching revision, a binary *has_template* label, which equals 0 for *negative* revisions, and 1 for *positive* revisions. We filter out templates with positive/negative pair counts of less than 1000. The remaining top 10 templates are shown in Table 1.

3.2 Downstream Tasks

We also identify potential *downstream tasks* that researchers in NLP and IR can perform on the data labeled with different templates, and assign them to each template in our dataset (see Table 1). We suggest two main tasks: *i*) Content Reliability Prediction (CRP) and *ii*) Source Retrieval (SR). The former consists in predicting the presence of a given template (the *has_template* column), while the latter consists in retrieving the sources and references that are

⁹https://figshare.com/articles/dataset/Wiki-Reliability_A_Large_Scale_Dataset_for_Content_Reliability_on_Wikipedia/14113799

Field	Description	Downstream Task
<i>page_id</i>	Page ID of the revision	all
<i>revision_id</i>	ID of the revision	all
<i>revision_id.key</i>	ID of the corresponding pos/neg revision	all
<i>txt_pos</i>	Wikipedia’s article in plain text when <i>has_template</i> is 1	all
<i>txt_neg</i>	Wikipedia’s article in plain text when <i>has_template</i> is 0	all
<i>revision_text_bytes</i>	Change in bytes of revision text	all
<i>stems_length</i>	Average length of stemmed text	all
<i>images_in_tags</i>	Count of images in tags	all
<i>infobox_templates</i>	Count of infobox templates	all
<i>paragraphs_without_refs</i>	Total length of paragraphs without references	all
<i>shortened_footnote_templates</i>	Number of shortened footnotes (i.e., citations with page numbers linking to the full citation for a source)	all
<i>words_to_watch_matches</i>	Count of matches from Wikipedia’s “words to watch”: words that are flattering, vague or endorsing a viewpoint	all
<i>revision_words</i>	Count of words for the revision	all
<i>revision_chars</i>	Number of characters in the full article	all
<i>revision_content_chars</i>	Number of characters in the content section of an article	all
<i>external_links</i>	Count of external links not in Wikipedia	all
<i>headings_by_level(2)</i>	Count of level-2 headings	all
<i>ref_tags</i>	Count of reference tags, indicating the presence of a citation	all
<i>revision_wikilinks</i>	Count of links to pages on Wikipedia	all
<i>article_quality_score</i>	Letter grade of article quality prediction	all
<i>cite_templates</i>	Count of templates that come up on a citation link	CRP
<i>who_templates</i>	Number of <i>who</i> templates, signaling vague “authorities”, i.e., “historians say”, “some researchers”	SR
<i>revision_templates</i>	Total count of all transcluded templates	SR
<i>category_links</i>	Count of categories an article has	SR
<i>has_template (label)</i>	Binary label indicating presence or absence of a reliability template in our dataset	all

Table 2: Schema of the Wiki-Reliability dataset, with the fields extracted from the positive/negative revision examples in our data. Information for each template is stored in two files, one with the text and another with all the additional features.

relevant to the article and improve the article quality. Note that although we provide a complete set of features plus the article content in plain text, additional information about references, links and categories can be extracted from the MediaWiki API¹⁰ using the page identifiers (see Table 2).

For example, revisions labeled with templates such as *Unreferenced* or *One Source*, can be used as a real-world testbed by researchers interested in applications around reference retrieval or recommendation. The data labeled with templates like *POV*, which indicates neutrality issues, or *Hoax*, reflecting potential falsehoods in the article, can be used to train classifiers able to identify content reliability issues such as the presence of fake, partisan or malicious information, similar to previous work detecting the citation needed template [10].

3.3 Metadata Features

We extract metadata features for each positive/negative revision in our data by querying the ORES API’s Article Quality model¹¹. The model predicts the article quality class of a Wikipedia article according to the Wikipedia article quality grading scheme¹². The model generates features based on the structural characteristics of an article and its revision. Features can be obtained with a scoring request¹³. We use the generated features along with the article quality score prediction as metadata features for our dataset, resulting in 26 metadata features in total. We classified them according to

the downstream task where they are used, removing features that would be artifact of the labels.

For our final dataset, we further narrow down the number of features to 20, by selecting only the most important features¹⁴ for template prediction tasks (see Section 4.2 for more details). The features released as part of the Wiki-Reliability dataset are listed in Table 2.

Figures 2a and 2b show the feature distribution for two features, *ref_tags* and *revision_chars*, across positive and negative revisions for different templates. We observe that, apart from *Unreliable Sources* and *Third Party*, articles revisions which have had their template removed, i.e. respective reliability issue resolved, tend to have more reference tags, suggesting that this metadata feature is actually informative of some reliability issues, as the feature distribution for positive revisions differ to the ones for negative revisions. Conversely, the number of characters in an article seems to be less informative of content reliability issues, since the *revision_chars* distribution tend to be similar across positive and negative examples for all templates.

3.4 Content Text

While some metadata features can be informative of the presence of a reliability template, researchers using this data might want to analyze the text of the revisions labeled as positive or negative. Models based on structural features might be unable to evaluate characteristics of the content such as writing quality, or the presence

¹⁰[urlhttps://www.mediawiki.org/wiki/API:Tutorial](https://www.mediawiki.org/wiki/API:Tutorial)

¹¹<https://www.mediawiki.org/wiki/ORES>

¹²https://en.wikipedia.org/wiki/Template:Grading_scheme

¹³https://www.mediawiki.org/wiki/ORES/Feature_injection

¹⁴To analyze the importance of different features, we averaged the importance scores of features from different content reliability prediction models, and removed the ones with lower significance score. Note that the models trained on the new reduced subset of features achieves comparable (and sometimes improved) accuracy to the full set.

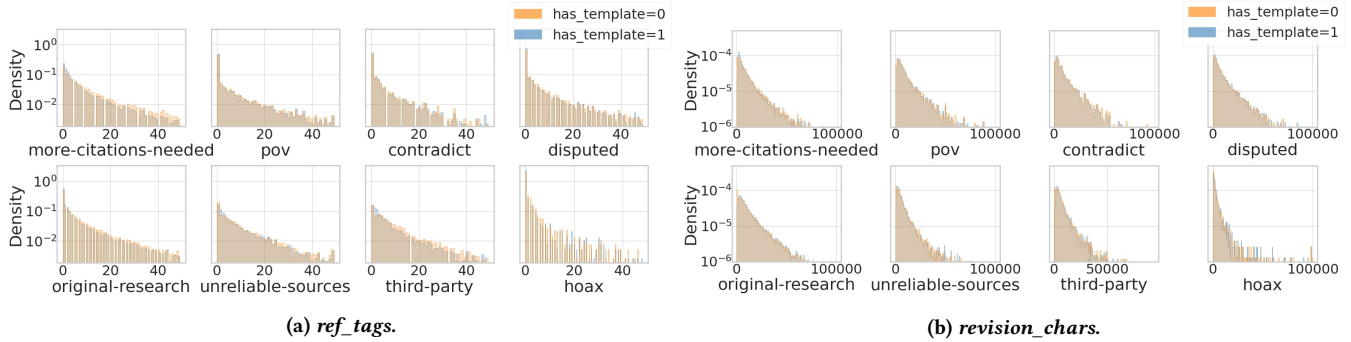


Figure 2: Distribution of the `ref_tags` (a) and `revision_chars` (b) features, reflecting respectively the number of reference tags in a revision and the length in characters of revision text, for positive and negative examples of each template.

of tonal issues, which might be more essential for the prediction of templates such as *POV* or *contradiction*. Thus, we also create a text-based dataset for the purpose of text classification.

For each revision in our dataset, we query the API for its *wikitext*, which we parse using `mwpaserfromhell` to obtain only the plain text content, stripping out all wikilinks, templates, and tags. Further, we filter out all references sections, keeping only the main article content.

4 METADATA-BASED MODELS AND ANALYSIS

To better understand the content and the informativeness of our dataset, we design a set of classifiers for the downstream task of content reliability prediction. For each template where we identified *CRP* as a potential downstream task (see Table 1), we learn a model predicting the target label `has_template` based on the metadata features.

4.1 Content Reliability Prediction Models

We train content reliability prediction models using `has_template` as the target variable, and the metadata features marked as *CRP* as the independent variables. For each template, we fit 3 models: Logistic Regression, Random Forest, and Gradient Boosted Trees.

Each classifier is trained using 3-fold cross validation, with a train/test split ratio of 2:1, using the `GroupKFold`¹⁵ iterator to enforce non-overlapping groups of articles across the training and test splits. This ensures that revisions of the same article will not appear in the test set if it already occurs in the training set, and vice versa, thus avoiding potential biases or overfitting. Training and test splits have balanced label distributions.

For all experiments, we report the average classification accuracy, precision, recall and F1-score over the 3 folds of the cross-validation.

4.2 Model Results

Across all templates, the Gradient Boost model achieves the highest performance scores. Figure 3 shows the accuracy score results for all templates: models perform better than random, with the maximum

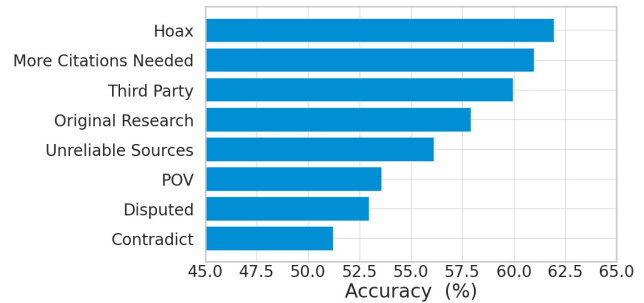


Figure 3: Average accuracy of the Gradient Boost-based prediction models for all predictable templates.

Template	Model	Accuracy	Precision	Recall	F1
Hoax	LogReg	0.58 ± 0.033	0.59 ± 0.033	0.58 ± 0.033	0.58 ± 0.035
	RF	0.58 ± 0.015	0.59 ± 0.015	0.58 ± 0.015	0.58 ± 0.014
	XGB	0.62 ± 0.014	0.62 ± 0.012	0.62 ± 0.014	0.62 ± 0.015
More Citations Needed	LogReg	0.56 ± 0.002	0.56 ± 0.004	0.56 ± 0.003	0.55 ± 0.003
	RF	0.58 ± 0.005	0.58 ± 0.005	0.58 ± 0.005	0.58 ± 0.006
	XGB	0.61 ± 0.000	0.61 ± 0.000	0.610 ± 0.000	0.61 ± 0.00
Third Party	LogReg	0.56 ± 0.005	0.58 ± 0.005	0.56 ± 0.005	0.56 ± 0.006
	RF	0.57 ± 0.006	0.578 ± 0.007	0.57 ± 0.006	0.57 ± 0.006
	XGB	0.60 ± 0.006	0.600 ± 0.005	0.60 ± 0.006	0.59 ± 0.006

Table 3: Metadata based models results for *Hoax*, *More Citations Needed* and *Third Party* templates: comparison between Logistic Regression (*LogReg*), Random Forests (*RF*), and Gradient Boosted Trees (*XGB*)

accuracy for template prediction standing at 62%. This suggests that the metadata features provided do carry some informativeness regarding the reliability of the articles' content. However, the relatively low performances of these models highlights the difficulty of this task, opening up opportunities for more research on this front.

Next, we look more in-depth at the top 3 templates by model accuracy: *Hoax*, *More Citations Needed* and *Third Party*. We present the full results of the classification performances for all metrics and methods in Table 3. These results confirm that, across all different metrics, the Boosting-based models perform better than others, with the *Hoax* template being more predictable than the other two.

¹⁵https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GroupKFold.html

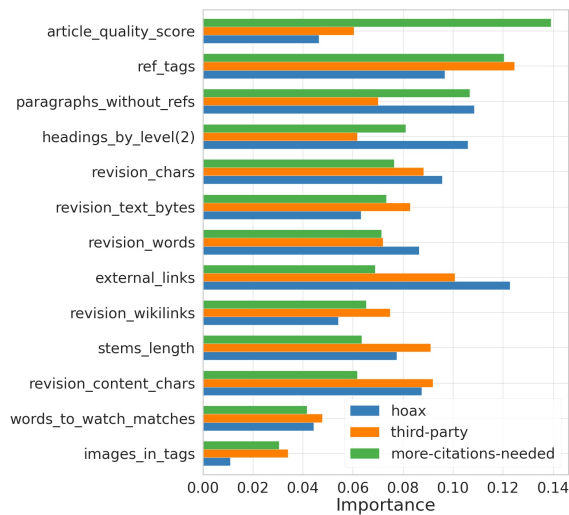


Figure 4: Feature importance for the models predicting the hoax, more citations needed, and third party templates.

To understand which features are most highly related with the presence of each template, we next obtain the feature importance scores computed from the Gradient Boosted Trees classifier. Feature importance is defined as the number of times a feature is used to split the data across all trees, divided by the total sum of occurrences. We compute these for the top 3 templates in Figure 4. We see that the presence of external links is the most significant feature for the *hoax* template prediction: by looking at the feature distribution, we see that the vast majority of articles tagged as “hoax” has indeed less than 2 external links, while pages where the “hoax” tags have been removed have a higher average number of external links. Article quality is a high predictor of the *more citations needed* template, probably due to the fact that the article quality model incorporates features such as the *ref_tags* variable, which reflects the number of references in an article and tends to have much lower values in presence of reliability templates. Finally, across all three templates we see that the length of paragraphs without references is an indicator of reliability issues. While these results suggest that metadata-based models have potentials for content reliability prediction, future research will need to use language models trained on textual content to improve detection results.

5 CONCLUSION AND FUTURE WORK

The *Wiki-Reliability* dataset contains Wikipedia articles labeled according to their reliability, using tags to signal specific reliability issues such as “non-neutral point of view”, or “disputed content”. We used the wisdom of experienced Wikipedia editors to label such articles, and additionally computed metadata features and release the articles’ full text. *Wiki-Reliability* is available for English Wikipedia only. Thanks to the reproducibility of our labeling method, we plan in the future to extend it to other languages.

We also suggested downstream tasks for this data. While these are just examples of what can be done with *Wiki-Reliability*, we hope that our proposed method for dataset creation, together with

the size and quality of the dataset generated, will foster creativity and promote research on novel NLP and IR tasks. We showed that classifiers trained on metadata features perform reasonably well for some downstream tasks, thus demonstrating the inherent quality of the data and the predictability of Wikipedia templates. To verify these observations, a manual evaluation of the quality of data will be performed as part of our future work.

Though training complex language models was outside the scope of this work, a simple logistic regression model trained on TF-IDF features gave promising results for the *CRP* task. This suggests that our data contains useful signals that a simple text-based model is able to capture, and we encourage researchers to use the large textual annotated data contained within this dataset to train more complex language models for content reliability tasks.

With this dataset, we intend to provide tools and data for researchers to build effective tools for automated support of Wikipedia content moderation, thus allowing the scientific community to contribute back to Wikipedia, and help to improve the quality of the main repository of human knowledge. While the main focus of this work is Wikipedia, content reliability systems trained on this data could be designed to generalize to contexts outside of Wikipedia, such as news articles or social media posts.

REFERENCES

- [1] Maik Anderka, Benno Stein, and Nedim Lipka. 2012. Predicting Quality Flaws in User-Generated Content: The Case of Wikipedia. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Portland, Oregon, USA) (SIGIR '12). Association for Computing Machinery, New York, NY, USA, 981–990. <https://doi.org/10.1145/2348283.2348413>
- [2] Ivan Beschastnikh, Travis Kriplean, and D. W. McDonald. 2008. Wikipedia Self-Governance in Action: Motivating the Policy Lens. In *International AAAI Conference on Web and Social Media (ICWSM)*.
- [3] Shruti Bhosale, Heath Vinicombe, and Raymond Mooney. 2013. Detecting Promotional Content in Wikipedia. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Seattle, Washington, USA, 1851–1857. <https://www.aclweb.org/anthology/D13-1190>
- [4] Marcel Forns, Joseph Allemandou, Dan Andreescu, and Nuria Ruiz. 2020. MediaWiki History: the best dataset on Wikimedia content and contributors. <https://techblog.wikimedia.org/2020/10/01/mediawiki-history-the-best-dataset-on-wikimedia-content-and-contributors/>
- [5] Andrea Forte, Vanesa Larco, and Amy Bruckman. 2009. Decentralization in Wikipedia Governance. *Journal of Management Information Systems* 26, 1 (2009), 49–72. <https://doi.org/10.2753/MIS0742-1222260103> [arXiv:https://doi.org/10.2753/MIS0742-1222260103](https://doi.org/10.2753/MIS0742-1222260103)
- [6] Aaron Halfaker and R. Stuart Geiger. 2020. ORES: Lowering Barriers with Participatory Machine Learning in Wikipedia. *Proc. ACM Hum.-Comput. Interact.* 4, CSCW2, Article 148 (Oct. 2020), 37 pages. <https://doi.org/10.1145/3415219>
- [7] Aniket Kittur, Bongwon Suh, Bryan A. Pendleton, and Ed H. Chi. 2007. He Says, She Says: Conflict and Coordination in Wikipedia. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*. Association for Computing Machinery, New York, NY, USA, 453–462. <https://doi.org/10.1145/1240624.1240698>
- [8] Florian Lemmerich, Diego Sáez-Trumper, Robert West, and Leila Zia. 2019. Why the world reads Wikipedia: Beyond English speakers. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 618–626.
- [9] Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2017. Advances in pre-training distributed word representations. *arXiv preprint arXiv:1712.09405* (2017).
- [10] Miriam Redi, Besnik Fetahu, Jonathan Morgan, and Dario Taraborelli. 2019. Citation needed: A taxonomy and algorithmic assessment of Wikipedia’s verifiability. In *The World Wide Web Conference*. 1567–1578.
- [11] Philipp Singer, Florian Lemmerich, Robert West, Leila Zia, Ellery Wulczyn, Markus Strohmaier, and Jure Leskovec. 2017. Why we read Wikipedia. In *Proceedings of the 26th international conference on world wide web*. 1591–1600.
- [12] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355* (2018).