With Few Eyes, All Hoaxes are Deep

SUMIT ASTHANA, IIT Patna, India

AARON HALFAKER, Wikimedia Foundation, USA

Quality control is critical to open production communities like Wikipedia. Editors enact quality control on the borders of Wikipedia to review edits (counter-vandalism) and new article creations (new page patrolling) shortly after they are saved. In this paper, we describe a long-standing set of inefficiencies that have plagued new page patrolling by drawing a contrast to the more efficient, distributed processes for counter-vandalism. To effect better page review distribution, we develop an effective automated topic model based on a labeling strategy that leverages a folksonomy developed by subject specific working groups in Wikipedia (*WikiProject* tags) and a flexible ontology (*WikiProjects Directory*) to arrive at a hierarchical and uniform label set. We are able to attain very high fitness measures (macro ROC-AUC: 95.2%, macro PR-AUC: 74.5%) and real-time performance using word2vec-based features on the initial draft versions of articles. Finally, we present a proposal for how incorporating this model into current tools will shift the dynamics of new article review positively.

CCS Concepts: • **Human-centered computing** → **Social recommendation**; **Computer supported cooperative work**; **Empirical studies in collaborative and social computing**; *Wikis*; *Social tagging systems*;

Keywords: Wikipedia; Social recommendation; Topic modeling; Collaborative Review

ACM Reference Format:

Sumit Asthana and Aaron Halfaker. 2018. With Few Eyes, All Hoaxes are Deep. In *Proceedings of the ACM on Human-Computer Interaction*, Vol. 2, CSCW, Article 21 (November 2018). ACM, New York, NY. 18 pages. <https://doi.org/10.1145/3274290>

1 INTRODUCTION

Wikipedia flips the traditional publication review process—publish first and ask questions later. Given the increasing article creation rate¹ on English Wikipedia, efficient review is the key for ensuring that only encyclopedic articles survive and inappropriate content is quickly removed. The easiest part of this review is to filter out articles that are outright vandalism, attack or spam. The more difficult part of this review is to take subtle judgment calls about the *notability* of the subject of an article. This latter review is plagued with a huge backlog. Previous studies suggest that the primary difficulty in reviewing this backlog involves dealing with such Time-Consuming Judgment Calls (TCJCs) [9, 18] - pages with potential that need work to bring them up to the current standards. We are motivated to reduce this difficulty of time-consuming judgment calls. In this paper, we describe the quality control dynamics in Wikipedia by highlighting the differences between new article review (an under-studied and poorly supported process) and new edit review (a highly studied and highly machine-supported process). In order to engineer towards new work dynamics,

¹<https://quarry.wmflabs.org/query/26385> shows 700-900 new articles per day

Authors' addresses: Sumit Asthana, IIT Patna, Patna, Bihar, 801103, India, asthana.sumit23@gmail.com; Aaron Halfaker, Wikimedia Foundation, Minneapolis, USA, ahalfaker@wikimedia.org.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

© 2018 Copyright held by the owner/author(s).

2573-0142/2018/11-ART21

<https://doi.org/10.1145/3274290>

Fig. 1. Left: New article review pipeline, Right: New edit review pipeline

we develop a machine prediction model, which is intended to lessen the difficulty of TCJCs by routing in expert reviewers sooner in the review process of a draft.

In order to build a model that is effective in routing expert Wikipedians, we leverage the folksonomies already in use in Wikipedia (WikiProjects and the WikiProject directory). This allows our modeling process to match Wikipedians' topic-focused working groups and to be flexible to future ontological repairs by Wikipedians. Specifically, *we develop a topic modeling strategy to predict topics for new Wikipedia drafts out of a set of 43 broad level categories specified by Wikipedians themselves.*

Finally, we discuss the potential for this model and provide a proposal to Wikipedians (and tool developers) to adopt it in order to reshape their review processes. While we build upon past work studying topic model using Wikipedia as a dataset, our focus is on describing and mitigating a specific collaborative workflow breakdown through a novel system design. We evaluate our system by showing that we can use it to effectively predict the topic of the initial version of articles—the focus of new page patrollers' work.

1.1 Quality control dynamics in Wikipedia

Currently, the processes that support reviewing new article creations in Wikipedia includes the New Page Patrol (supported by the Page Curation Tool² (NPP) and the Articles for Creation³ (AfC)) working group. The only substantial difference between these two processes from a workflow perspective is where they operate and the implications of that space. NPP operates in the main encyclopedia article space and thus is motivated to make sure problematic new page creations don't last long because they show up in front of readers. AfC operates in the Draft namespace where new articles are not linked from the main encyclopedia and are not indexed by search engines.

While AfC and NPP are somewhat independent, they largely work in parallel as a single line of defense against the introduction of spam, vandalism, and other types of non-encyclopedic content. In the case of both AfC and NPP, a small group of volunteers is responsible for making a wide range of different types of judgement calls about a new article. Some judgments are straightforward and require little expertise: *E.g.*, is this spam or vandalism? Others require a nuanced understanding of a topic area: *E.g.*, is Ann Bishop a notable biologist or a hoax?

Historically, new article review has been an overwhelming problem[18]. Both AfC and NPP maintain backlogs with tens of thousands of articles⁴. Recently, NPP has become so overwhelmed with the backlog and the potentially negative effects that it has been having on the quality of

²<https://enwp.org/WP:NPP>

³<https://enwp.org/WP:AfC>

⁴https://en.wikipedia.org/wiki/?oldid=828015939#Backlog_11,800

the encyclopedia that patrollers have proposed that Wikipedia change policy and restrict the creation of new articles from new editors⁵. This has largely had the effect of re-routing new article review work from NPP (main space) to AfC (draft space), but it has not addressed the backlog at all[23]. Worse, recent studies have shown that routing newcomers to this hidden draft space is counter-productive [19] because it reduces opportunities for collaboration.

However, not all of Wikipedia's open review processes result in such backlogs. It's interesting to compare the review process for edits to current articles (edit review) with the review process funnel for new articles (Fig. 1). In the case of edit review, a dynamic multi-stage filter is implemented based on a distributed cognition process[6]. First, automated bots review edits using AIs to detect obvious damage, then AI-augmented human-computation tools catch less obvious damage, then finally, editors from across the encyclopedia review changes to articles that they're interested via watchlists[5]. Through this multi-stage process, all edits are reviewed efficiently with increasingly complex judgement calls being made later in the process. Because this system is generally considered to be robust, there's no backlog that is maintained of unreviewed edits. Figure 1 shows a visual comparison of the levels of review between new articles (one level) and edits to existing articles (three levels). See Geiger and Halfaker[5] for a discussion of the temporal rhythms of edit review.

Our goal in this work is to use this insight to develop a solution to tackle the limitations of new article review process in Wikipedia. Our work is focused on making it possible to implement AI-augmented multi-stage review processes for new article review that lessens the burden of TCJCs on current patrollers by routing new pages to subject matter experts. We maintain that we need machine support to tag articles with topics rather than having the authors tag the topics because 1) newcomers would be inconsistent with tagging and their tags would likely not match Wikipedia's existing ontology, 2) we have a huge backlog of untagged drafts that needs addressal, 3) with a model, changes to the tagging ontology can be automatically propagated to all drafts.

The paper is divided as follows: We start by discussing about related works in Artificial Intelligence around Wikipedia aimed at assisting the editor community. We also include previous work on multilabel classification, our prime mode of classification here. Then we move on to describe the drafttopic dataset and detail the steps on how we generated it using our open source drafttopic api and our novel WikiProjects parser. We also argue the flexibility around the dataset. Then we describe our experiments using this new dataset to come up with a topic prediction model, describing the features - word-vectors and bag-of-words and classifiers used by us. Lastly we put forth our baseline results of the best experiments and some interesting observations on predictions with this model around drafts on English Wikipedia.

2 RELATED WORK

Wikipedia has been using Artificial Intelligence to support various operations that help make collaboration easier. A crucial one of these is the use of classifiers to counter-vandalism on Wikipedia edits [13],[16]—by highlighting edits that are likely to be damaging, the workload of reviewing can be dramatically lessened. Other work has developed article quality prediction models⁶ based on *actionable* article features[24]. These models are used in task routing⁷ and to help students write better Wikipedia articles[15]. Recently, the Scoring Platform team at the Wikimedia Foundation released "draftquality"⁸ model that is designed to detect "vandalism", "spam", and "personal attack" articles for quick deletion. Predictions of all of these models can be accessed by the ORES[8] webservice that the Wikimedia Foundation provides via public apis. We are only beginning to see

⁵<https://en.wikipedia.org/wiki/Wikipedia:ACTRIAL>

⁶e.g., <https://github.com/wiki-ai/wikiclass>

⁷e.g., <https://enwp.org/User:DataflowBot> uses quality prediction to highlight high potential articles

⁸<https://github.com/wiki-ai/draftquality>

the immense benefits of making general machine learning available to support quality management workflows.

Based on past work studying the article review workflows in Wikipedia[18, 23], it is clear that these review processes are struggling while others are robust and manageable[5]. Based on recommendations in past work, it seems clear that the primary gap between edit review and article review is the potential of task routing. For new edit review, editors can employ "watchlists"⁹ to tag articles that they are interested in and be notified of changes to those articles. This allows editors to review changes to articles that they are interested in—topics where TCJC's are simply easier. There's no such alternative that exists for new article creations since an editor can't "watch" an article before it is created. Therefore, we seek to find a signal for the topic space of new article creations such that interested editors can *watch* new page creations too. This is in line with design recommendations by Schneider et al.[18]

To realize such task routing, we take the approach to predict topic labels for articles based on Wikipedian's own topic-focused working groups: WikiProjects. There has been extensive work around topic modeling in Wikipedia that suggests we can find success. Blei et al.[1] showed how we can get a set of important words for each document denoting its topical composition using LDA modeling. LDA is an effective topic modeling strategy but it infers topics from the corpus automatically where each topic is a composition of words from the corpus in different proportions. Here, we can use WikiProject categories as labeled data. Labeled-LDA[14] is one such work that allows forcing a set of topic labels on the LDA algorithm. However, our experiments with labeled LDA were unsuccessful because the current implementations of the algorithm¹⁰ work with dense matrices which are not computationally efficient (*i.e.*, unmanageable in production). Therefore to better place our work in line with previous works, we use the standard One-Vs-All classification with SVMs and bag-of-words to establish a baseline that roughly matches past work with topic models in Wikipedia. This simple approach has yielded robust results in the past[22] to compare newer, more complicated approaches against, on standard multilabel datasets like RCV1[10] or yahoo science datasets.

Word embeddings¹¹ have shown promise as an effective feature engineering strategy for extracting signal from text[21]. Compared to a simple bag-of-words strategy, word embeddings allow for very efficient compression of signal. Essentially this means it can be used to build a memory efficient prediction model with high fitness. Further, Li et al. showed that such word embeddings can be an effective strategy for topic modeling very short texts[11]—like article drafts that tend to consist of 2-5 sentences. Based on these insights, we develop a word2vec¹² based averaging strategy to build a model that is fast to train and can be efficiently deployed in production. We leave the use of more advanced, but less performant Labeled-LDA, and LSTMs as interesting future works on this dataset.

To the best of our knowledge, we are the first to intervene in this difficult workflow with a viable technological task routing strategy. But our work is part of a long line of studies exploring how article creation[18, 23] and reviews work in Wikipedia's peer production processes[5].

3 BUILDING THE TOPIC DATASET

In order to develop a high-level topic labeling of Wikipedia, we explored many options. Generally, we wanted to predict a relatively small set of topics (less than 100) in order to make sure our model was small enough to work in practice. Further these topics should cover the space of Wikipedia

⁹<https://enwp.org/Special:Watchlist>

¹⁰<https://nlp.stanford.edu/software/tmt/tmt-0.4/>

¹¹https://en.wikipedia.org/wiki/Word_embedding

¹²<https://en.wikipedia.org/wiki/Word2vec>

subjects uniformly. We briefly considered using the category system built into Wikipedia. While Wikipedia does have an extensive category network[25], we abandoned that idea early because of lack of a proper hierarchy among the categories. Another option seemed to provide great potential. WikiProjects are subject-focused working groups – the kind of subject-interested working groups we wanted to target with our topic modeling. These working groups tag articles that fall within the scope of their projects. E.g. WikiProject Birds tags the article for Eagle as within their content space. Regretfully there are hundreds of WikiProjects and the coverage of WikiProjects is not very uniform. E.g. some projects are extremely broad (E.g. WikiProject Asia) while others are more focused (E.g. Chinese military history). Each WikiProject has an associated template by the name "WikiProject:X" which can be added to an article's talk page, signifying the article's membership in that WikiProject topic "X".

The WikiProjects Directory¹³ provides a convenient intermediate ontology of WikiProjects that starts with four broad topics: Culture, Geography, History & Society and Science, Technology and Mathematics (STEM) (Fig. 2). From there, the directory drills down into sub-topics and eventually specific WikiProjects. For example, *WikiProject Birds* exists underneath the path STEM/Science/Animals.

We utilized the well defined hierarchy of the directory pages to group WikiProjects under 43 different *mid level categories* which formed our prediction space. These *mid level categories* are nothing but 2nd level headings prefixed by their first level parent in the hierarchy given on the WikiProjects directory home page. For example, *WikiProject:Television* can be looked up from the *Broadcasting* entry(2nd level) in the tree under *Culture and the Arts*(1st level) hence its grouped under the mid level category *Culture.Broadcasting*(First_level.second_level). Fig. 3 shows a snapshot of the culture section of the directory home page.

By taking advantage of this directory structure and using the mid-level categories and the WikiProject taggings already available on Wikipedia, we can (1) have a small, uniform set of target class and (2) enable editors to control the class output. Imagine a scenario where something isn't quite right with the WikiProject Directory structure and so it doesn't capture the right level of detail in order to be useful. E.g. editors want to target biographies about women and yet "women" does not show up at any level of the directory. An editor could add a new level under Culture/Biography called "Women" and include WikiProject Women Writers, WikiProject Women Scientists, and other women-focused WikiProjects underneath this new branch of the tree. We can then run the same automated scripts for extracting structure on the augmented tree to re-label articles using these new mid-level categories without wasting any further human-effort.

In this way, the directory *acts as a layer of ontological abstraction between the WikiProject taggings and the uniform set of categories that we want to use in order to model topics on Wikipedia* (Fig. 2). By allowing volunteer editors to easily modify this ontology and implementing automated systems for re-interpreting the ontology, we enable a straightforward process for adjusting our prediction model to match their expectations. We suspect that our prediction model will highlight limitations in the current directory structure and that editors will be able to work with us in a tight loop of iterations to converge on an optimal set of target labels for the model.

3.0.1 Possible biases. Given that our drafttopic dataset's foundation is the Wikiproject's directory, which is created by Wikipedians themselves, its definitely subject to human biases. A bias in the editing patterns of the encyclopedia would reflect in the taxonomy and it would automatically result in a bias in the topic distribution of the dataset. For example, if we were to train our model before the Women Scientist coverage gap was filled [7], the model would likely have formed a negative association between content about women and STEM topics. While we have no means of

¹³https://enwp.org/WP:WikiProject_Council/Directory

Fig. 2. The WikiProjects directory abstraction and relation to mid-level categories

Culture and the arts

- Arts
 - Music
 - Performing
 - Plastic
 - Visual
- Broadcasting
- Crafts and hobbies
- Entertainment
 - Games and toys
- Food and drink
- Internet culture
- Language and literature

Music [edit source]

Project	Active	Assessment	Peer review	Collaboration	Portal	Notes
Music	yes				Music	Guidelines: Manual of Style (music)
→					Music	
→					Music	
→					Music	
Music terminology	no				Music	
Music theory	yes	yes			Music	
Theme songs	no					Currently listed as defunct
Tunings, Temperaments and Scales	no					Currently listed as defunct

Composers [edit source]

Project	Active	Assessment	Peer review	Collaboration	Portal	Notes
Composers	yes	yes			Classical music	
Gilbert and Sullivan						(main listing here)
Richard Wagner	yes	yes			Opera	

Compositions [edit source]

Project	Active	Assessment	Peer review	Collaboration	Portal	Notes
Songs	yes	yes			Music	

Fig. 3. Left: Snapshot of culture section of WikiProjects directory, Right: Snapshot of Music table linked to from the directory page

knowing all possible biases beforehand, the model itself highlights the biases in the taxonomy and this may provide an opportunity for Wikipedians to identify the underlying causes and mitigate them. By basing this workflow support strategy on a model that can be re-trained and re-deployed, we make it easier for such adjustments to be made.

3.1 Machine-readable dataset

The first step in generating the topic dataset is to parse the user generated WikiProjects directory information into a machine readable json format. We achieved this by writing a WikiProjects parser that starts from the directory home page and recursively iterates over each entry and traverses over any subpage that the entry links to, doing the same thing again in the new subpage. Fig. 3 shows such a subpage linked to from the directory page by the *Music* entry. Note the well structured tables

containing WikiProjects. The parser preserves the hierarchy of the way WikiProjects are organized. Every new entry that it sees on the main page or a subpage which is NOT a WikiProject entry, is made an internal node under which subsequent entries are grouped which that entry links to. Any WikiProject entry is made a leaf node in the parsing, a base case for the recursion to stop further parsing. We use regexes to identify whether an entry is a WikiProject entry or an intermediate subtopic (internal node) requiring further expansion.

3.2 Regexes

Our extraction process used of regexes to parse tables, sections, directory pages and following the links. The wikitext that we parse looks like:

```
{{Wikipedia:WikiProject Council/Directory/WikiProject\n
|project = ([a-zA-Z_-: -]+\n
|shortname = ([a-zA-Z\(\) -]+\n
|active = (yes|no)\n([^\}]*))}}
```

ALGORITHM 1: Create Machine Readable WikiProjects json

```
processed = {};
function parse_wp_directory():
    wp_json = {};
    for entry in directory tree do
        processed[entry] = True;
        subpage = FETCH_PAGE(entry);
        wp_json[entry] = GET_SUB_CATEGORIES(wp_json, subpage);
    end
    return wp_json;
function get_sub_categories(wp_json, subpage):
    wp_json = {};
    for sections in subpage do
        wp_json[section] = {};
        processed[section] = True;
        if not processed[section] then
            | wp_json[section] = GET_SUB_CATEGORIES(wp_json, section)
        end
    end
    return wp_json
```

3.3 Mid-level categories

The first step gives us the machine readable version of the WikiProjects topic hierarchy. However, this still doesn't contain explicitly the topics we intend to predict. As discussed before, our motive is to predict the broad level categories present on the directory home page. Thus we prune the machine readable topic tree by traversing it in a depth first manner and stopping at a certain depth of the tree and aggregating all leaf nodes under that node as directly belonging to the last traversed node. We chose to target 2nd level entries present on the directory home page. For example, *WikiProject:Nicklodeon* can be followed as *Culuture and the Arts* → *Broadcasting* → *Television* → *American TV shows* → *WikiProject:Nicklodeon*. However, since we prune our traversal at *Broadcasting* it comes directly under *Broadcasting*. At the end of this step, we finally have the

topic labels we intend to predict mapped to the individual WikiProjects they correspond to, which actually exist in article talk pages. E.g. the entry *Culture.Language and literature* maps to the WikiProjects - *WikiProject Linguistics*, *WikiProject Writing systems*, *WikiProject Academic Journals*, etc.. (Fig. 2)

3.4 Extracting article topic labels

We used the above dataset which contains forty three different mid-level-categories each containing different WikiProjects within them to get article-topic labels. We used PAWS¹⁴ for this purpose which is a Wikimedia service for running python code on Wikimedia cluster that can directly interact with WMF databases.

Some topics had many more articles than others. To ensure that our classifier had enough observations to learn each class, we built a stratified sample by extracting 2000 article ids per mid-level category. To do this, we matched the WikiProjects covered by a mid-level category to the *templatelinks* table to identify WikiProject tags.

At the end of this step, we have a set of approximately 93000 article entries each of which had a *page-id*, *mid-level categories* and associated WikiProject *templates*. Note that we did not get full 96000 articles as some mid-level categories contained very few articles. Due to overlap between mid-level categories (e.g. Ann Bishop appears in both STEM.Biology and Culture.Language and literature), some categories were represented more strongly than others in our sample so we were not able to attain perfect balance. However, this method ensured that all categories were at least minimally represented in order to train and test against.

4 METHODS

Overall, our pipeline is divided into three stages 1) Fetching data, 2) Feature engineering and preprocessing, 3) Classification

4.1 Fetching Article Data

This step involves querying the Wikipedia api to fetch the first version of article text for the set of *page_ids* that are specified in the WikiProjects dataset. We extract the raw article text using a *fetch_text* extraction api from our code. All features are extracted from the text of this initial revision so as to represent the *draft* version of the article—the target of our routing strategy.

4.2 Feature Engineering

We used word2vec^[12] and bag-of-words as features for our two different approaches. Word vectors already provide a lot of contextual information in a dense representation and save on space and processing time. This is essential for our use case since we intend for this model to be used in real-time by Wikipedia editors to support their work. We used 300-dimensional Google-News word vectors with negative sampling as our features. For extracting the word vectors we made use of a revscoring¹⁵ extension to access gensim¹⁶ that provides an easy to use interface to load the Goolge-News binary and extract vectors.

4.2.1 Feature Extraction. To extract features from the fetched data, the revscoring library requires specifying a features file that dictates how to extract the features as a set of *dependents*¹⁷. We defined this pipeline by creating a simple word2vec feature extraction file that loads the word2vec binary,

¹⁴<https://paws.wmflabs.org/>

¹⁵<https://pythonhosted.com/revscoring>

¹⁶<https://radimrehurek.com/gensim/models/word2vec.html>

¹⁷https://github.com/wiki-ai/revscoring/blob/master/ipynb/feature_engineering.ipynb

defines a filter to extract only non-stop words (*english.stopwords.revision.datasources.non_stopwords*), and returns the averaged vector of all non-stopwords in the article passed to it. This feature file is used by the feature extraction utility *extract_from_text* that extracts features for each observation and rewrites the dataset augmented with the *cache* attribute containing the extracted features. For bag-of-words, we used the tf-idf vectorizer from scikit-learn and extracted tf-idf features from each article's first version.

4.3 Classification

We used linear SVMs[2] for our bag-of-words approach. For word vectors we fed them directly to Random Forests and Gradient Boosting classifiers. We chose these classifiers because they have been found to be robust and fast in previous applications of AI to wiki processes around the ORES machine learning service[17]. Applications like *edit quality* and *article quality* predictions in Wikipedia have shown promising results with these memory-efficient estimators.

Mathematically speaking, let X denote the instance space and $\mathcal{L} = \{\lambda_1, \lambda_2, \dots, \lambda_n\}$ be a finite set of labels. Each instance is associated with a subset of labels $\ell \in 2^{\mathcal{L}}$. This subset is called relevant labels and the complement is called the set of irrelevant labels. We identify a set L of relevant labels with the binary vector $\mathbf{y} = \{y_1, y_2, \dots, y_m\}$ in which $y_i = 1 \Leftrightarrow \lambda_i \in L$. By $\mathcal{Y} = \{0, 1\}^m$ we denote the set of all possible labelings. Revscoring internally uses scikit-learn's classifiers training and predictions methods, but uses its own implementation for evaluating predictions. We tuned hyperparameters through the use of a simple grid-search strategy.

4.3.1 Random Forests. Random Forests[3] are a collection of tree based classifiers that help to reduce variance by training a number of independent tree based classifiers. We provided explicit label weights to the Random Forests since the training set per label is highly unbalanced. This is expected because each article is only expected to be related to a subset of topics. Typically the ratio of pre-label positive to negative instances in the dataset is 1:10 for each label. For assigning the weight to each instance we simply assign the weight based on its positive to negative samples ratio.

We use the following hyperparameters set for RandomForest:

- n_estimators - 10, 20, 40, 80, 160, 320
- max_depth - 3, 4
- max_features - sqrt, log

4.3.2 Gradient Boosting Classifiers. We also used Gradient Boosted Trees [4] because boosting is a proven method to tackle both variance and bias.

The following hyperparameters set for GradientBoostingClassifiers was used:

- n_estimators - 10, 20, 40, 80, 160, 320
- max_depth - 3, 4, 5, 6
- max_features - sqrt, log
- learning_rate - 0.01, 0.1, 1, 10

4.3.3 Gradient Boosting Classifiers. Linear SVMs were used for the tf-idf features. The following was the hyperparameters set:

- C - 1, 10, 100
- kernel - Linear

4.4 Training

We used virtual machines on Wikimedia servers for training all of our models. It was a Debian server with 36 GB of RAM and 8 CPUs. For all training and testing purposes the revscoring library is optimized to use maximum available CPUs for parallelizing work.

5 EVALUATION

Since our approach is binary relevance, which builds a separate classifier for each target label, the evaluation metric is just extending the binary metrics for each label and aggregating. We calculated the micro precision, recall and ROC-AUC for each label using 5-fold cross-validation and then aggregate them as macro precision, macro recall, and macro ROC-AUC.

5.1 Metric

let X denote the evaluation dataset consisting of $|X|$ multilabel instances (x_i, Y_i) , $i = 1 \dots |X|$, $Y_i \in \mathcal{L}$, the label set. Let H be the multilabel classifier and $Z_i = H(x_i)$ be the set of labels predicted by H for example x_i . We use the following:

$$Accuracy(H, X) = \frac{1}{X} \sum_{i=1}^{|X|} \frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|}$$

$$Precision(H, X) = \frac{1}{X} \sum_{i=1}^{|X|} \frac{|Y_i \cap Z_i|}{|Z_i|}$$

$$Recall(H, X) = \frac{1}{X} \sum_{i=1}^{|X|} \frac{|Y_i \cap Z_i|}{|Y_i|}$$

Tables 1 and 2 show the micro and macros statistics respectively for the two models used by us. We omit mentioning the precision, recall values as they are threshold dependent but mention the ROC-AUC and PR-AUC which are threshold agnostic and better representative of the performance of classification.

	Accuracy	PR-AUC	ROC-AUC
RandomForests	91.9	64.6	93.9
GradientBoosting	97.5	74.5	95.2
SVM	97.4	76.41	85.9

Table 1. Macro Statistics for RandomForest, Gradient Boosting and SVM models

	Accuracy	PR_AUC	ROC_AUC
RandomForests	87.9	66.2	91.9
GradientBoosting	94.5	77.8	94.9
SVM	94.1	76.41	85.69

Table 2. Micro Statistics for RandomForest, Gradient Boosting and SVM models

Figures 4 and 5 show the precision-recall and ROC curve for the gradient boosting classifier. For visual clarity reasons, we chose select categories, but made sure to include one mid-level category from each section *STEM*, *Geography*, *Culture and History*. For all metrics, the Gradient Boosting classifier with word vectors and SVM with bag-of-words were comparable and they performed slightly better than RandomForest classifiers.

Fig. 4. PR curve for Gradient Boosting Model

Fig. 5. ROC curve for Gradient Boosting Model

5.2 Best features and scalability considerations

The best performance was achieved with the following hyper parameters set:

- RandomForest - *n_estimators*: 320, *max_depth*: 4, *max_features*: log2
- GradientBoostedTrees - *n_estimators*: 320, *max_depth*: 5, *max_features*: log2, *learning_rate*: 0.1
- SVM - *Kernel*: Linear, *C*: 1

We note that although the best fitness was with $n_estimators$: 320, the model suffered from a high prediction time cost (approximately 6s per instance). Since prediction time and estimators are linearly related, using 100 estimators per classifier brought us down to 2s limit without a substantial compromise in fitness. Figure 6 gives the dependence of PR-AUC on max-depth and n-estimators and validates our claim. Since being practical was also one of the objectives of this model, 100 estimators provides good enough efficiency in time without compromising much on fitness. Moreover, the 300 features word-vector based models were far faster to train and test than bag-of-words with SVM.

Fig. 6. Variation of PR-AUC with max-depth and n-estimators

It's also worth noting that our balanced dataset used in training and cross-validation does not match the target use-case of the model since draft articles will be randomly distributed across the target mid-level categories. So we performed a follow-up test on a random sample of 50K articles that have been tagged by at least one WikiProject. On this dataset our selected Gradient Boosting model scored a micro PR-AUC of 0.824 and ROC-AUC of 0.834. This result suggests that the model will perform effectively across the natural topic distribution of draft articles.

6 ERROR ANALYSIS

While we maintain that the WikiProject judgements assigned to articles by users are trustworthy since they have gone through Wikipedia's rigorous review process, we also identify that they are not complete. This is almost identifiable from the confusion matrix (Appendix) where the classifier had a large number of false positives but very few false negatives. A lot of the false positives are due to the fact that the classifier predicted correct topics based on presence of topic matter where editors had not yet tagged articles appropriately. We should probably not consider these to be false positives – just gaps in existing information. For example, the article about Ann Bishop (biologist) does not include a WikiProject tag for "Biology" and thus is not labeled as STEM. Biology even though it clearly should be! This observation about missing labels suggests that the fitness measurements are generally conservative. It also suggests that there is a surprising utility to our topic model: It can be used to predict missing topics for existing full-fledged articles!

6.1 Examples of missing labels

Some of the predictions inspected by us reveal the truth in the above statement regarding predicting missing topics as well as some mistakes performed by the classifier.

The article *Interleukin-7_receptor* is a clear example of our model being able to predict gaps in existing tags. The article clearly is about a chemical compound with some information of the diseases associated with it. However it was only tagged with *STEM.Biology* and *STEM.Medicine* while our model also predicts *STEM.Chemistry* which appears to be the primary topic of the article.

The article *360s* is interesting in the sense that it lists all the events that took place in 360 AD with short descriptions. The actual tag for the article is *STEM.Time* but our model also predicts *History_And_Society.History and society*, *Geography.Europe*, *History_And_Society.Politics and government*. Looking at the article we can see that the predictions are relevant as the article discusses history with a lot of geo-political content around the Roman Empire.

The article *Cavitation*, about formation of vapour cavities in liquids is tagged with *STEM.Technology*, *STEM.Physics* by the users but a quick glance of the table of contents shows that it has medical, chemical, engineering and geo-scientific applications. Our model also predicted *STEM.Engineering*, *STEM.Chemistry*, *STEM.Medicine*, *STEM.Geosciences*, *STEM.Science*.

Fig. 7. Predictions for the biologist Ann Bishop and true labels

Ann Bishop (Fig. 7), a notable biologist, fell under *Culture.Language & Literature*, *History.History&Society* but did not have any STEM category. Our model correctly predicted *STEM.Medicine* and *STEM.Biology* in addition to the above, which is her field of profession.

7 FUTURE WORK AND VISION

We envision that this model will change the dynamics around new article review on Wikipedia. Our hope is based on our observations that we described at the start and the effectiveness with which our model predicts topics for articles. Geiger et. al[5] talks about the importance of algorithmic support in Wikipedia's edit review system and the adverse effects of its absence for short periods on edit review. Just like that, we argue that incorporating this model's predictions into ORES and its live use by Wikipedia editors in the Wikipedia new article review interface, will go a long way in aiding these reviewers in judging the content space of new articles and reducing the Time Consuming Judgement Calls discussed in the beginning. Once our model starts providing topic-specific signal for new articles, tool developers can experiment with strategies for routing new article creations to the right experts where time consuming judgement calls are less difficult. It occurs to us in this work that we're applying the insights of Linus' law¹⁸: "given enough eyeballs, all bugs are shallow". This saying about open-source software suggests that given enough beta-testers and bug-reporters, all issues will be discovered and reported. In the case of new article review, the lack of topic routing failed to attract many eyes to the problem of reviewing articles in the way that it attracts many eyes to the problem of reviewing edits. Past work has made it clear that this is a problem for new

¹⁸https://enwp.org/Linus%27s_Law

article review[18], and now with our modeling work, steps can be taken to route new articles to reviewers with the relevant topic interest/expertise.

The existing draftquality model and the drafttopic model we describe are expected to work in tandem—the first filtering out outright inappropriate drafts and the latter helping to route the good one’s to appropriate content spaces thereby ensuring easier review of less concerning content (Fig. 8). Through the application of these models to pre-filter and route reviews, we hope to enable the new page reviewing process to look more like the edit review process (see Fig. 1). In addition to aiding wiki processes, we also hope that the topic modeling corpus generated by us containing a rich set of Wikipedia articles tagged with *mid_level_categories* will be useful in other domains

Fig. 8. The drafttopic and draftquality models helping editors filter out "OK" drafts and then routing them to topic namespaces

8 APPLICABILITY TO OTHER SOCIAL COMPUTING SYSTEMS

We have demonstrated an approach here that makes use of implicit feedback in a collaborative social system and uses that to build models that aid in processes around that system. The general methodology of starting from a consistent collaborative space (the WikiProjects directory) and building up the dataset using a combination of database queries and scripts and then using that in the revscoring library to train models could well be extended to other social domains which employ implicit tagging or judgments. We hope that our methods of pruning the topic tree to arrive at an intermediate representation of WikiProject topics that are tunable by the community, testing and analysis would prove useful for the community when extending the work to other domains. Another important aspect where our work can help the collaborative learning community is *filling content gaps in existing social systems* as discussed in *Error analysis*. By definition, social systems are collaborative in nature and there will be content gaps based on the demographics of collaborators which can be addressed by such strategies as ours.

In this case, we leveraged a collaboratively curated directory structure (WikiProject Directory) on top of a folksonomy (WikiProject tags) to implement a high fitness, memory efficient classifier (using word embeddings), for routing content review. Other peer-production communities are beginning to struggle with the efficient quality control as they reach scales nearing Wikipedias[20]. We hope that this work will pave the way for other communities to address their new content review backlogs more efficiently by leveraging the distributed expertise of their local crowd.

Lastly, we hope that the multilabel dataset we created and its analysis would be useful for the topic modeling community in general for algorithmic improvements.

9 CONCLUSION

In this paper, we describe a specific work context in Wikipedia that is struggling. Wikipedia, the open encyclopedia is less open today because the new article reviewers are overwhelmed. Through examining new article reviewer processes and comparing them to the more-efficient new edit review processes we were able to identify a critical gap: topic routing. In response, we built a topic model using an ontology and folksonomy developed and maintained by Wikipedians for organizing content by subject interest (WikiProjects). By combining this folksonomy and a set of already tagged articles, we were able to build a `mid_level_categories` mapping for training a prediction model. We used `word2vec` based embeddings to develop a powerful and efficient strategy for feature extraction of topic signal. Finally, we tested Random Forests, Gradient Boosting classifiers and SVMs to perform multilabel classification on this dataset and achieved a high level of fitness with both bag-of-words and word vectors.

These models are fast and easy to train and use which makes them practical for deployment in a production/real-time environment like that of ORES, a machine prediction service that is already widely used by Wikipedians. Finally, we have proposed how our model and other models that are already deployed in production can be used to re-shape the new article review process in productive ways.

Given the high level of fitness we were able to attain with basic binary classification methods, we leave the application of more advanced label dependence methods as future work. Further we encourage the readers to make use of our WikiProjects ontological abstractions and extend them to other domains for topic modeling. We suspect that the strict hierarchy of the WikiProjects directory along with the WikiProjects tagging can provide a very useful view into Wikipedia's topic spaces where other research has struggled to use categories. Such an ontology could be very beneficial for ongoing work exploring Wikipedia's topic coverage bias like that covered by Halfaker[7].

APPENDIX

Code

We have provided the code for this work on github in *drafttopic*¹⁹ part of the AI repositories for Wikimedia. We have provided both the datasets - the high level hierarchy of WikiProject topics (*wp_2017-11-05.json*) and the labeled WikiProjects set containing 93445 article topic mappings (*enwiki.labeled_wikiprojects.json*). These are present in the datasets folder and the *README* specifies how to reproduce them. The steps to build a model are enumerated in the section *Train a gradient boosting model*.

Confusion Matrix

Table 3 shows the confusion matrix for each label generated by the gradient boosting classifier. The true positive and false positives vary greatly across categories because of the relative popularity of the categories and the ability of the classifier to generalize. The given confusion matrix is for the parameter set *n_estimators: 100, max_depth: 3, max_features=log2, learning_rate=0.1*. The categories *Assistance.Files*, *Assistance.Maintenance*, *Assistance.Content Systems* and *Assistance.grading* are for administrative purposes and can be ignored for now.

¹⁹<https://github.com/wikimedia/drafttopic>

	TruePositive	FalsePositive	FalseNegative	TrueNegative
STEM.Time	1515 (1.62%)	867 (0.93%)	116 (0.12%)	90917 (97.33%)
STEM.Physics	1498 (1.60%)	1135 (1.22%)	378 (0.40%)	90404 (96.78%)
STEM.Space	2135 (2.29%)	387 (0.41%)	101 (0.11%)	90792 (97.19%)
STEM.Mathematics	1090 (1.17%)	569 (0.61%)	74 (0.08%)	91682 (98.14%)
Culture.Crafts and hobbies	1236 (1.32%)	914 (0.98%)	67 (0.07%)	91198 (97.63%)
Geography.Transportation	3091 (3.31%)	1185 (1.27%)	339 (0.36%)	88800 (95.06%)
Geography.Maps	1374 (1.47%)	1178 (1.26%)	73 (0.08%)	90790 (97.19%)
Culture.Internet culture	1295 (1.39%)	855 (0.92%)	266 (0.28%)	90999 (97.41%)
H&S.Business and economics	3525 (3.77%)	3058 (3.27%)	863 (0.92%)	85969 (92.03%)
Assistance.grading	9 (0.01%)	62 (0.07%)	223 (0.24%)	93121 (99.69%)
STEM.Biology	1944 (2.08%)	1493 (1.60%)	291 (0.31%)	89687 (96.01%)
STEM.Science	630 (0.67%)	1680 (1.80%)	138 (0.15%)	90967 (97.38%)
STEM.Technology	1715 (1.84%)	2568 (2.75%)	692 (0.74%)	88440 (94.67%)
Geography.Countries	15173 (16.24%)	10403 (11.14%)	4533 (4.85%)	63306 (67.77%)
Culture.Media	703 (0.75%)	1503 (1.61%)	253 (0.27%)	90956 (97.37%)
H&S.History and society	1677 (1.80%)	5905 (6.32%)	633 (0.68%)	85200 (91.21%)
Assistance.Contents systems	465 (0.50%)	1618 (1.73%)	284 (0.30%)	91048 (97.47%)
Culture.Broadcasting	2124 (2.27%)	1198 (1.28%)	419 (0.45%)	89674 (96.00%)
Assistance.Maintenance	1835 (1.96%)	4557 (4.88%)	437 (0.47%)	86586 (92.69%)
Culture.Food and drink	1746 (1.87%)	726 (0.78%)	168 (0.18%)	90775 (97.17%)
H&S.Military and warfare	2952 (3.16%)	1443 (1.54%)	394 (0.42%)	88626 (94.87%)
Culture.Philosophy and religion	1998 (2.14%)	2118 (2.27%)	489 (0.52%)	88810 (95.07%)
Geography.Oceania	2594 (2.78%)	1618 (1.73%)	197 (0.21%)	89006 (95.28%)
Culture.Plastic arts	2343 (2.51%)	1581 (1.69%)	438 (0.47%)	89053 (95.33%)
H&S.Education	1769 (1.89%)	1012 (1.08%)	245 (0.26%)	90389 (96.76%)
STEM.Engineering	852 (0.91%)	1526 (1.63%)	289 (0.31%)	90748 (97.14%)
Culture.Visual arts	2960 (3.17%)	1911 (2.05%)	519 (0.56%)	88025 (94.23%)
STEM.Meteorology	1725 (1.85%)	393 (0.42%)	66 (0.07%)	91231 (97.66%)
Culture.Language and literature	14935 (15.99%)	5592 (5.99%)	2269 (2.43%)	70619 (75.60%)
Geography.Landforms	1731 (1.85%)	498 (0.53%)	133 (0.14%)	91053 (97.47%)
Culture.Entertainment	4336 (4.64%)	1849 (1.98%)	608 (0.65%)	86622 (92.73%)
STEM.Medicine	1645 (1.76%)	868 (0.93%)	294 (0.31%)	90608 (97.00%)
Geography.Bodies of water	1829 (1.96%)	507 (0.54%)	157 (0.17%)	90922 (97.33%)
STEM.Information science	1274 (1.36%)	862 (0.92%)	130 (0.14%)	91149 (97.57%)
Culture.Sports	4150 (4.44%)	877 (0.94%)	370 (0.40%)	88018 (94.22%)
Geography.Europe	9081 (9.72%)	7021 (7.52%)	2590 (2.77%)	74723 (79.99%)
H&S.Politics and government	1830 (1.96%)	2541 (2.72%)	609 (0.65%)	88435 (94.67%)
Geography.Cities	438 (0.47%)	383 (0.41%)	117 (0.13%)	92477 (99.00%)
STEM.Geosciences	1281 (1.37%)	804 (0.86%)	140 (0.15%)	91190 (97.62%)
Culture.Performing arts	3878 (4.15%)	843 (0.90%)	436 (0.47%)	88258 (94.48%)
Culture.Arts	1563 (1.67%)	516 (0.55%)	89 (0.10%)	91247 (97.68%)
STEM.Chemistry	1760 (1.88%)	759 (0.81%)	308 (0.33%)	90588 (96.97%)
Assistance.Files	16 (0.02%)	358 (0.38%)	47 (0.05%)	92994 (99.55%)

Table 3. Confusion matrix for the GradientBoosting model, H&S-History And Society

REFERENCES

- [1] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.
- [2] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. 1992. A Training Algorithm for Optimal Margin Classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory (COLT '92)*. ACM, New York, NY, USA, 144–152. <https://doi.org/10.1145/130385.130401>
- [3] Leo Breiman. 2001. Random Forests. *Mach. Learn.* 45, 1 (Oct. 2001), 5–32. <https://doi.org/10.1023/A:1010933404324>
- [4] Jerome H. Friedman. 2000. Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics* 29 (2000), 1189–1232.
- [5] R. Stuart Geiger and Aaron Halfaker. 2013. When the Levee Breaks: Without Bots, What Happens to Wikipedia’s Quality Control Processes?. In *Proceedings of the 9th International Symposium on Open Collaboration (WikiSym '13)*. ACM, New York, NY, USA, Article 6, 6 pages. <https://doi.org/10.1145/2491055.2491061>
- [6] R. Stuart Geiger and David Ribes. 2010. The work of sustaining order in wikipedia: the banning of a vandal. In *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work, CSCW 2010, Savannah, Georgia, USA, February 6-10, 2010*, Kori Inkpen Quinn, Carl Gutwin, and John C. Tang (Eds.). ACM, 117–126. <https://doi.org/10.1145/1718918.1718941>
- [7] Aaron Halfaker. 2017. Interpolating Quality Dynamics in Wikipedia and Demonstrating the Keilana Effect. In *Proceedings of the 13th International Symposium on Open Collaboration, OpenSym 2017, Galway, Ireland, August 23-25, 2017*, Lorraine Morgan (Ed.). ACM, 19:1–19:9. <https://doi.org/10.1145/3125433.3125475>
- [8] Aaron Halfaker, R. Stuart Geiger, Jonathan Morgan, Amir Sarabadani, and Adam Wight. 2018. Topic modeling for short texts with auxiliary word embeddings. (2018). https://commons.wikimedia.org/wiki/File:ORES_-_Facilitating_re-mediation_of_Wikipedia%27s_socio-technical_problems.pdf
- [9] Danny Horn. 2017. New pages patrol: Analysis and proposal. https://en.wikipedia.org/wiki/Wikipedia:New_pages_patrol/Analysis_and_proposal
- [10] David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. 2004. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research* 5, Apr (2004), 361–397.
- [11] Chenliang Li, Haoran Wang, Zhiqian Zhang, Aixin Sun, and Zongyang Ma. 2016. Topic modeling for short texts with auxiliary word embeddings. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 165–174.
- [12] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *CoRR abs/1301.3781* (2013). arXiv:1301.3781 <http://arxiv.org/abs/1301.3781>
- [13] Martin Potthast, Benno Stein, and Robert Gerling. 2008. Automatic vandalism detection in Wikipedia. In *European conference on information retrieval*. Springer, 663–668.
- [14] Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D Manning. 2009. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*. Association for Computational Linguistics, 248–256.
- [15] Sage Ross. 2016. Visualizing article history with structural completeness. <https://wikiedu.org/blog/2016/09/16/visualizing-article-history-with-structural-completeness/>
- [16] Amir Sarabadani, Aaron Halfaker, and Dario Taraborelli. 2017. Building Automated Vandalism Detection Tools for Wikidata. In *Proceedings of the 26th International Conference on World Wide Web Companion, Perth, Australia, April 3-7, 2017*, Rick Barrett, Rick Cummings, Eugene Agichtein, and Evgeniy Gabrilovich (Eds.). ACM, 1647–1654. <https://doi.org/10.1145/3041021.3053366>
- [17] Amir Sarabadani, Aaron Halfaker, and Dario Taraborelli. 2017. Building automated vandalism detection tools for Wikidata. In *Proceedings of the 26th International Conference on World Wide Web Companion*. International World Wide Web Conferences Steering Committee, 1647–1654.
- [18] Jodi Schneider, Bluma S. Gelley, and Aaron Halfaker. 2014. Accept, Decline, Postpone: How Newcomer Productivity is Reduced in English Wikipedia by Pre-publication Review. In *Proceedings of The International Symposium on Open Collaboration (OpenSym '14)*. ACM, New York, NY, USA, Article 26, 10 pages. <https://doi.org/10.1145/2641580.2641614>
- [19] Jodi Schneider, Bluma S. Gelley, and Aaron Halfaker. 2014. Accept, decline, postpone: How newcomer productivity is reduced in English Wikipedia by pre-publication review. In *OpenSym*. ACM, 26:1–26:10.
- [20] Nathan TeBlunthuis, Aaron Shaw, and Benjamin Mako Hill. 2018. Revisiting The Rise and Decline in a Population of Peer Production Projects. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 355.
- [21] Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*. Association for Computational Linguistics, 384–394.
- [22] Naonori Ueda and Kazumi Saito. 2003. Parametric mixture models for multi-labeled text. In *Advances in neural information processing systems*. 737–744.

- [23] Morten Warncke Wang. 2018. Autoconfirmed Article Creation Trial. https://meta.wikimedia.org/wiki/Research:Autoconfirmed_article_creation_trial
- [24] Morten Warncke-Wang, Dan Cosley, and John Riedl. 2013. Tell me more: an actionable quality model for Wikipedia. In *Proceedings of the 9th International Symposium on Open Collaboration, Hong Kong, China, August 05 - 07, 2013*, Ademar Aguiar and Dirk Riehle (Eds.). ACM, 8:1–8:10. <https://doi.org/10.1145/2491055.2491063>
- [25] Torsten Zesch and Iryna Gurevych. 2007. Analysis of the Wikipedia category graph for NLP applications. In *Proceedings of the Second Workshop on TextGraphs: Graph-Based Algorithms for Natural Language Processing*. 1–8.

Received April 2018; revised July 2018; accepted September 2018