Growing Wikipedia Across Languages via Recommendation

Ellery Wulczyn
Wikimedia Foundation
ellery@wikimedia.org

Robert West
Stanford University
west@cs.stanford.edu

Leila Zia
Wikimedia Foundation
leila@wikimedia.org

Jure Leskovec
Stanford University
jure@cs.stanford.edu

ABSTRACT

The different Wikipedia language editions vary dramatically in how comprehensive they are. As a result, most language editions contain only a small fraction of the sum of information that exists across all Wikipedias. In this paper, we present an approach to filling gaps in article coverage across different Wikipedia editions. Our main contribution is an end-to-end system for recommending articles for creation that exist in one language but are missing in another. The system involves identifying missing articles, ranking the missing articles according to their importance, and recommending important missing articles to editors based on their interests. We empirically validate our models in a controlled experiment involving 12,000 French Wikipedia editors. We find that personalizing recommendations increases editor engagement by a factor of two. Moreover, recommending articles increases their chance of being created by a factor of 3.2. Finally, articles created as a result of our recommendations are of comparable quality to organically created articles. Overall, our system leads to more engaged editors and faster growth of Wikipedia with no effect on its quality.

1. INTRODUCTION

General encyclopedias are collections of information from all branches of knowledge. Wikipedia is the most prominent online encyclopedia, providing content via free access. Although the website is available in 291 languages, the amount of content in different languages differs significantly. While a dozen languages have more than one million articles, more than 80% of Wikipedia language editions have fewer than one hundred thousand articles [24]. It is fair to say that one of the most important challenges for Wikipedia is increasing the coverage of content across different languages.

Overcoming this challenge is no simple task for Wikipedia volunteers. For many editors, it is difficult to find important missing articles, especially if they are newly registered and do not have years of experience with creating Wikipedia content. Wikipedians have made efforts to take stock of missing articles via collections of "redlinks"[1] or tools such as "Not in the other language" [15]. Both technologies help editors find missing articles, but leave editors with long, unranked lists of articles to choose from. Since editing Wikipedia is unpaid volunteer work, it should be easier for editors to find articles missing in their language that they would like to contribute to. One approach to helping editors in this process is to generate personalized recommendations for the creation of important missing articles in their areas of interest.

Although Wikipedia volunteers have sought to increase the content coverage in different languages, research on identifying missing content and recommending such content to editors based on their interests is scarce. Wikipedia's SuggestBot [6] is the only end-to-end system designed for task recommendations. However, SuggestBot focuses on recommending existing articles that require improvement and does not consider the problem of recommending articles that do not yet exist.

Here we introduce an empirically tested end-to-end system to bridge gaps in coverage across Wikipedia language editions. Our system has several steps: First, we harness the Wikipedia knowledge graph to identify articles that exist in a source language but not in a target language. We then rank these missing articles by importance. We do so by accurately predicting the potential future page view count of the missing articles. Finally, we recommend missing articles to editors in the target language based on their interests. In particular, we find an optimal matching between editors and missing articles, ensuring that each article is recommended only once, that editors receive multiple recommendations to choose from, and that articles are recommended to the the most interested editors.

We validated our system by conducting a randomized experiment, in which we sent article creation recommendations to 12,000 French Wikipedia editors. We find that our method of personalizing recommendations doubles the rate of editor engagement. More important, our recommendation system increased the baseline article creation rate by a factor of 3.2. Also, articles created via our recommendations are of comparable quality to organically created articles. We conclude that our system can lead to more engaged editors and faster growth of Wikipedia with no effect on its quality.

The rest of this paper is organized as follows. In Sec. 2 we present the system for identifying, ranking, and recommending missing articles to Wikipedia editors. In Sec. 3 we describe how we evaluate each of the three system components. In Sec. 4 we discuss the details of the large scale email experiment in French Wikipedia. We discuss some of the opportunities and challenges of this work and some future directions in Sec. 5 and share some concluding remarks in Sec. 6.

2. SYSTEM FOR RECOMMENDING MISSING ARTICLES

We assume we are given a language pair consisting of a *source language S* and a *target language T*. Our goal is to support the creation of important articles missing in *T* but existing in *S*.

[1] Redlinks are hyperlinks that link from an existing article to a non-existing article that should be created.

WWW 2016, April 11–15, 2016, Montréal, Québec, Canada.
ACM 978-1-4503-4143-1/16/04.
http://dx.doi.org/10.1145/2872427.2883077.

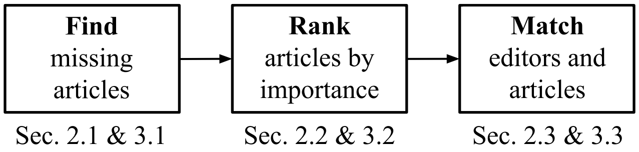

Find missing articles
Rank articles by importance
Match editors and articles
Sec. 2.1 & 3.1
Sec. 2.2 & 3.2
Sec. 2.3 & 3.3

Figure 1: System overview. Sec. 2.1, 2.2, 2.3 describe the components in detail; we evaluate them in Sec. 3.1, 3.2, 3.3.

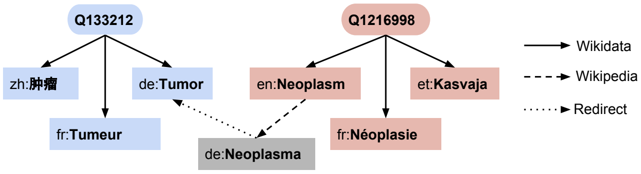


Figure 2: Language-independent Wikidata concepts (oval) linking to language-specific Wikipedia articles (rectangular). Clusters are merged via redirects and inter-language links hand-coded by Wikipedia editors.

Our system for addressing this task consists of three distinct stages (Fig. 1). First, we find articles missing from the target language but existing in the source language. Second, we rank the set of articles missing in the target language by importance, by building a machine-learned model that predicts the number of views an article would receive in the target language if it existed. Third, we match missing articles to well-suited editors to create those articles, based on similarities between the content of the missing articles and of the articles previously edited by the editors. The steps are explained in more details in the following sections.

2.1 Finding missing articles

For any pair of languages (S,T), we want to find the set of articles in the *source language* S that have no corresponding article in the *target language* T. We solve this task by leveraging the *Wikidata* knowledge base [20], which defines a mapping between language-independent concepts and language-specific Wikipedia articles. For example, the abstract concept Q133212 in Wikidata maps to 62 language-specific Wikipedia articles about tumors (such as the TUMOR article in German Wikipedia, TUMEUR in French Wikipedia or NOVOTVORINA in Croatian Wikipedia). We refer to these language-independent concepts as *Wikidata concepts*.

This mapping induces a clustering of the Wikipedia articles from all languages, such that each cluster contains all articles about the same concept in the different languages. Therefore, a simple approach to finding articles that are present in S but missing in T would be to consider those concepts whose cluster contains an article in S but none in T. We could, for example, assume that Estonian Wikipedia has no coverage of the TUMOR concept because the corresponding Wikidata concept Q133212 has no link to the Estonian language.

A complicating factor is that distinct Wikidata concepts may correspond to nearly identical real-world concepts, but every Wikidata concept can link to only one article per language. For example, there are separate Wikidata concepts for NEOPLASM and TUMOR. The English and German Wikipedias have decided that these concepts are similar enough that each only covers one of them: German covers TUMOR, while English covers NEOPLASM, so the simple approach described above would consider the NEOPLASM article to be missing in German—something we want to avoid, since the topic is already covered in the TUMOR article.

In order to solve this problem, we need a way to partition the set of Wikidata concepts into groups of near-synonyms. Once we have such a partitioning, we may define a concept c to be missing in language T if c's group contains no article in T.

In order to group Wikidata concepts that are semantically nearly identical, we leverage two signals. First, we extract inter-language links which Wikipedia editors use to override the mapping specified by Wikidata and to directly link articles across languages (*e.g.*, in Fig. 2, English NEOPLASM is linked via an inter-language link to German NEOPLASMA). We only consider inter-language links added between S and T since using links from all languages has been shown to lead to large clusters of non-synonymous concepts [3]. Second, we extract intra-language redirects. Editors have the ability to create redirects pointing to other articles in the same language (*e.g.*, as shown in Fig. 2, German Wikipedia contains a redirect from NEOPLASMA to TUMOR). We use these two additional types of links—inter-language links and redirects—to merge the original concept clusters defined by Wikidata, as illustrated by Fig. 2, where articles in the red and blue cluster are merged under the same concept by virtue of these links.

In a nutshell, we find missing articles by inspecting a graph whose nodes are language-independent Wikidata concepts and language-specific articles, and whose edges are Wikidata's concept-to-article links together with the two additional kinds of link just described. Given this graph, we define that a concept c is missing in language T if c's weakly connected component contains no article in T.

2.2 Ranking missing articles

Not all missing articles should be created in all languages: some may not be of encyclopedic value in the cultural context of the target language. In the most extreme case, such articles might be deleted shortly after being created. We do not want to encourage the creation of such articles, but instead want to focus on articles that would fill an important knowledge gap.

A first idea would be to use the curated list of 10,000 articles every Wikipedia should have [23]. This list provides a set of articles that are guaranteed to fill an important knowledge gap in any Wikipedia in which they are missing. However, it is not the case that, conversely, all Wikipedias would be complete if they contained all of these articles. For instance, an article on PICADA, an essential aspect of Catalan cuisine, may be crucial for the Catalan and Spanish Wikipedias, but not for Hindi Wikipedia. Hence, instead of trying to develop a more exhaustive global ranking and prioritizing the creation of missing content according to this global ranking, we build a separate ranking for each language pair.

Ranking criterion. We intend to rank missing articles according to the following criterion: *How much would the article be read if it were to be created in the given target language?* Since this quantity is unknown, we need to predict it from data that is already known, such as the popularity of the article in languages in which it already exists, or the topics of the article in the source language (all features are listed below).

In particular, we build a regression model for predicting the normalized rank (with respect to page view counts) of the article in question among all articles in the target language. The normalized rank of concept c in language T is defined as

$$y_T(c) := \frac{\text{rank}_T(c)}{|T|}, \tag{1}$$

where $\mathrm{rank}_T(c)$ is the (unnormalized) rank of T's article about concept c when all articles in language T are sorted in increasing order of page view counts. We considered page views received by each article in our dataset over the period of six months prior to the data collection point. Page view data was obtained via the raw HTTP request logs collected by the Wikimedia Foundation. Requests from clients whose user-agent string reveals them as bots were excluded [14].

For a given source–target pair (S,T), the model is trained on concepts that exist in both languages, and applied on concepts that exist in S but not in T. We experiment with several regression techniques (Sec. 3.2), finding that random forests [5] perform best.

Features. Finally, we describe the features used in the regression model for source language S and target language T. Here, c_L is the article about concept c in language L. Information about c_T is not available at test time, so it is also excluded during training.

Wikidata count: The more Wikipedias cover c, the more important it is likely to be in T. Hence we include the number of Wikipedias having an article about c.

Page views: If c is popular in other languages it is also likely to be popular in T. Thus these features specify the number of page views the articles corresponding to c have received over the last six months in the top 50 language versions of Wikipedia. Since some languages might be better predictors for T than others, we include page view counts for each of the 50 languages as a separate feature. (If the article does not exist in a language, the respective count is set to zero.) In addition to the raw number of page views, we also include the logarithm as well as the normalized page view rank (Eq. 1).

Geo page views: If c_S is popular in certain countries (presumably those where T is spoken), we expect c_T to be popular as well. Hence these features specify the number of page views c_S has received from each country.

Source-article length: If c_S contains substantial content we expect c to be important in T. Hence we consider the length of c_S (measured in terms of bytes in wiki markup).

Quality and importance classes: If c_S is considered of high quality or importance, we expect c to be an important subject in general. To capture this, we use two signals. First, several Wikipedias classify articles in terms of quality as ‘stub’, ‘good article’, or ‘featured article’ based on editor review [25]. Second, members of WikiProjects, groups of contributors who want to work together as a team to improve a specific topic area of Wikipedia, assign importance classes to articles to indicate how important the article is to their topical area [22]. We compute the maximum importance class that c_S has been given by any WikiProject. Quality and importance class labels are coded as indicator variables.

Edit activity: The more editors have worked on c_S, the more important we expect c to be for T as well. Hence we consider the number of editors who have contributed to c_S since it was created, as well as the number of months since the first and last times c_S was edited.

Links: If c_S is connected to many articles that also exist in T then the topical area of c is relevant to T. Therefore this feature counts the numbers of inlinks (outlinks) that c_S has from (to) articles that exist in T. We also include the total indegree and outdegree of c_S.

Topics: Some topics are more relevant to a given language than others. To be able to model this fact, we build a topic model over all articles in S via Latent Dirichlet Allocation (LDA) [4] and include the topic vector of c_S as a feature. We use the LDA implementation included in the *gensim* library [18], set the number of topics to 400, and normalize all topic vectors to unit length.

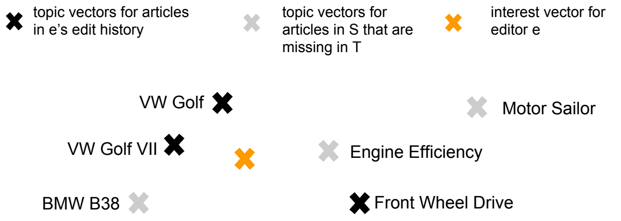

✖ topic vectors for articles in e's edit history
✖ topic vectors for articles in S that are missing in T
✖ interest vector for editor e
VW Golf
VW Golf VII
BMW B38
Motor Sailor
Engine Efficiency
Front Wheel Drive

Figure 3: Embedding of an editor's edit history into the topic vector space.

2.3 Matching editors to articles

Our high-level objective is to encourage editors to create important articles. We have already described how to find important missing articles (Sec. 2.1, 2.2). What is needed next is to find the best editors to create those articles. We hypothesize that editors are more likely to create articles that fall into their area of interest, and therefore we need a way of capturing how interested an editor is in creating a given article (Sec. 2.3.1). Finally, in order to make the most effective recommendations, we need a way to combine the inherent importance of an article with an editor's interest in creating that article (Sec. 2.3.2), and to match editors with articles based on the resulting scores (Sec. 2.3.3).

2.3.1 Editor interest modeling

For an editor e and concept c we want to score how closely c matches the topics of interest to e. Later we will use these interest scores to match editors with missing articles.

We quantify e's interest in creating an article about c via the similarity of c to the articles e has previously edited (*i.e.*, e's *edit history*). This idea is illustrated schematically in Fig. 3, where the black crosses represent the articles e has edited, summarized in a single point by the orange cross. Gray dots stand for missing articles that could potentially be recommended to e. The closer two points, the more similar the concepts they represent, and the closer a missing article (gray cross) is to e's summarized edit history (orange cross), the better a recommendation it is. Operationalizing this idea poses the challenges of (1) representing articles in a vector space and (2) aggregating an entire edit history into a single point in that vector space.

Vector-space representation of concepts. First, to embed concepts in vector space, we represent the concept c by the LDA topic vector of c_S (*cf.* Sec. 2.2). We can include contributions e made in languages other than S if the edited articles have a corresponding article in S. In this case, we use the topic vector of c_S to represent the edited article. The distance between two concepts is measured as the Euclidean[2] distance between their normalized topic vectors.

Aggregating edit histories into interest vectors. Second, to summarize edit histories as interest vectors, we proceed as follows. For each revision made by e, we compute the number of bytes that were added to the article. We then compute the total number of bytes e has added to an article over the course of all the revisions to that article. Revisions that remove bytes are not included. This way, each concept appears at most once in each edit history. We consider three different methods of summarizing an edit history into a single vector in the LDA topic space, which we refer to as the editor's *interest vector* (all interest vectors are normalized to unit length):

[2] Since vectors are normalized, cosine distance and Euclidean distance are equivalent for our purposes.

1. **Average.** The interest vector for editor e is computed as the mean of the topic vectors of all articles in e's edit history.
2. **Weighted average.** As above, with the difference that each concept c is weighted by the logarithm of the number of bytes e has added to c.
3. **Weighted medoid.** The interest vector is defined as the topic vector of the article from e's edit history that minimizes the weighted sum of distances between it and all other topic vectors from the edit history. Weights are computed as above.

History size. When computing an interest vector, we may not want to include all articles from the edit history. For instance, some editors have worked on thousands of articles, which leads to unnecessarily long interest vector computation times and very dense interest vectors when using averaging. An editor's interests may also evolve over time, so including articles edited a long time ago adds noise to the signal of what topics the editor is interested in now. For these reasons, we introduce *history size* as a tunable parameter, specifying the number of most recently edited articles considered for computing interest vectors.

2.3.2 Integrating importance and interest

We aim to recommend a missing article c_T to an editor e if (1) concept c is important in the target language T and (2) it is relevant to e's interests. Above, we have proposed methods for quantifying these two aspects, but in order to make effective recommendations, we need to somehow integrate them.

A simple way of doing so is to first rank all articles missing from language T by importance (Sec. 2.1), then discard all but the K most important ones (where K is a parameter), and finally score the relevance of each remaining concept c for editor e by computing the distance of c's topic vector and e's interest vector (Sec. 2.3.1).

A slightly more complex approach would be to keep all missing articles on the table and compute a combined score that integrates the two separate scores for article importance and editor–article interest, *e.g.*, in a weighted sum or product.

Both approaches have one parameter: the number K of most important articles, or the weight for trading off article importance and editor–article interest. Since we found it more straightforward to manually choose the first kind of parameter, we focus on the sequential strategy described first.

2.3.3 Matching

When several editors simultaneously work on the same article, the danger of edit conflicts arises. In order to prevent this from happening, we need to ensure that, at any given time, each article is recommended to only a single editor. Further, to avoid overwhelming editors with work, we can make only a limited number of recommendations per editor. Finally, we want the articles that we recommend to be as relevant to the editor as possible. Formally, these goals are simultaneously met by finding a matching between editors and articles that maximizes the average interest score between editors and assigned articles, under the constraints that each article is assigned to a unique editor and each editor is assigned a small number k of unique articles.

We formulate this matching problem as a linear program [7] and solve it using standard optimization software.

In practice, we find that a simple greedy heuristic algorithm gives results that are as good as the optimal solutions obtained from the matching algorithm. The heuristic algorithm iterates k times over the set of editors for whom we are generating recommendations and assigns them the article in which they are interested most and which has not yet been assigned.

Rank	Lenient precision		Strict precision	
1	0.85	[0.64, 0.95]	0.55	[0.34, 0.74]
101	0.90	[0.70, 0.97]	0.55	[0.34, 0.74]
1,001	0.95	[0.76, 0.99]	0.90	[0.70, 0.97]
10,001	1.00	[0.84, 1.00]	0.95	[0.76, 0.99]
100,001	1.00	[0.84, 1.00]	0.95	[0.76, 0.99]

Table 1: Empirical values with 95% credible intervals for precision of missing-article detector (Sec. 2.1). Rows are predicted importance levels of missing articles; for definition (also of two kinds of precision), *cf.* Sec. 3.1.

3. OFFLINE EVALUATION

Before evaluating our system based on deploying the complete pipeline in a live experiment (Sec. 4), we evaluate each component offline, using English as the source language S, and French as the target language T.

3.1 Finding missing articles

Here we assess the quality of the procedure for detecting missing articles (Sec. 2.1). We do not assess recall since that requires a ground-truth set of missing articles, which we do not have. Furthermore, using English as a source and French as the target, our procedure produces 3.7M missing articles. Given this large number of articles predicted to be missing, we are more concerned with precision than with recall. Precision, on the other hand, is straightforward to evaluate by manually checking a sample of articles predicted to be missing for whether they are actually missing.

Our first approach to evaluating precision was to sample 100 articles uniformly at random from among the 300K most important ones (according to our classifier from Sec. 2.2). This gives 99% precision: only one of 100 actually exists. The English ECTOSYMBIOSIS was labeled as missing in French, although French ECTOSYMBIOSE exists—it just has not been linked to Wikidata yet. Since precision might not be as high for the most important missing articles, we ran a second evaluation. Instead of taking sample test cases at random, we took a sample stratified by predicted importance of the missing article. We manually checked 100 cases in total, corresponding to ranks 1–20, 101–120, 1,001–1,020, 10,001–10,020, and 100,001–100,020 in the ranking produced by our importance model (Sec. 2.2).

In addition to the unambiguous errors arising from missing Wikidata and inter-language links, we observed two types of cases that are more difficult to evaluate. First, one language might spread content over several articles, while the other gathers it in one. As an example, French covers the Roman hero HERCULE in the article about the Greek version HÉRACLÈS, while English has separate articles for HERCULES and HERACLES. As a consequence, the English article HERCULES is labeled as missing in French. Whether HERCULE deserves his own article in French, too, cannot be answered definitively. Second, a concept might currently be covered not in a separate article but as a section in another article. Again, whether the concept deserves a more elaborate discussion in an article of its own, is a subjective decision that has to be made by the human editor. Since it is hard to label such borderline cases unequivocally, we compute two types of precisions. If we label these cases as errors, we obtain *strict precision;* otherwise we obtain *lenient precision*.

Precision results are summarized in Table 1. While we overall achieve good performance (lenient precision is at least 85% across importance levels), the results confirm our expectation that detecting missing articles is harder for more prominent concepts (strict

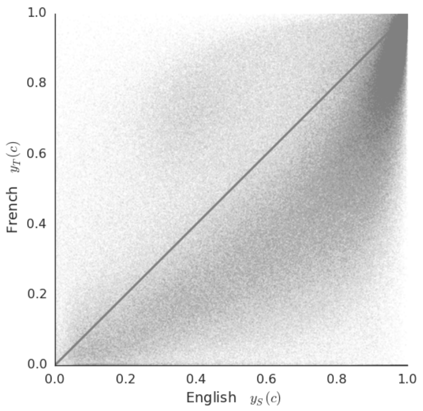


Figure 4: Scatterplot of page view ranks for articles in English *vs.* French Wikipedia.

precision is only 55% for the two highest importance levels). We conclude that it is important to ask editors to whom we make suggestions to double-check whether the article is really missing in the target language. Since the effort of this manual check is small compared to the effort of creating the article, this is a reasonable requirement.

3.2 Ranking missing articles

In this section we discuss the performance of our method for ranking missing articles by importance. Recall that here importance is measured in terms of the number of page views received by a page once it is created. We use English as the source, and French as the target language.

Evaluation metrics. We use two evaluation metrics, Spearman's rank correlation coefficient and root mean squared error (RMSE). The former is appropriate as we care mostly about the ranking of predictions, rather than the exact predicted values. The latter is useful because it has a natural interpretation: since we predict normalized page view ranks (Sec. 2.2), an RMSE of, say, 0.1 means that the average article ranks 10 percentile points higher or lower than predicted.

Baselines. The simplest baseline is to always predict the same constant value. We use the mean normalized rank over all articles in the target language (*i.e.*, 0.5) as the constant and call this the *mean baseline*.

It is reasonable to assume that the newly created version of an article in the target language will not be too different in terms of page view rank, compared to the version in the source language. Hence our second baseline (termed *source-language baseline*) is given by the normalized rank of the already existing source version of c ($y_S(c)$ in the notation of Eq. 1).

Random forests. In order to improve upon these baselines, we experimented with several regression techniques, including linear regression, ridge regression, least-angle regression, and random forests, using implementations in *scikit-learn* [17]. We found that random forests [5] gave the best performance, so we focus on them. In all experiments, the data is split into a training and a testing set. To tune the hyperparameters of the random forest model (the number of trees in the ensemble, as well as their maximum depth), we perform cross-validation on the training set.

Model	RMSE	Spearman correlation
Mean baseline	0.287	N/A
Source-language baseline	0.276	0.673
Random forests	0.130	0.898

Table 2: Importance ranking test results (Sec. 3.2).

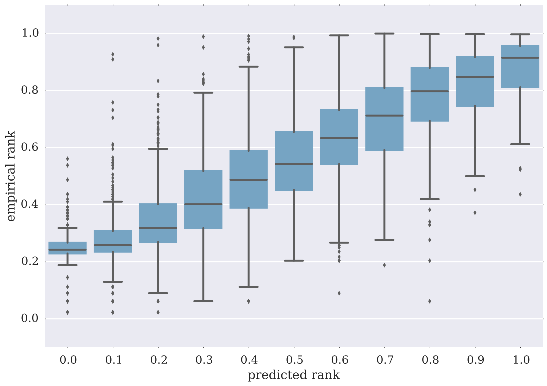


Figure 5: Empirical page view ranks for newly created articles as a function of predicted ranks (for target language French).

Results. As seen in Table 2, the simple mean baseline yields an RMSE of 0.287. The source-language baseline improves on this only slightly, with an RMSE of 0.276. Fig. 4 plots the prediction of this baseline against the ground-truth value (*i.e.*, it shows a scatter plot of the normalized ranks in English and French). We see that, while there is significant correlation, the source-language rank tends to overestimate the target-language rank.

Table 2 compares the performance of the baselines with the tuned random forest regression model. The latter performs better by a large margin in terms of both RMSE (0.130 *vs.* 0.276) and Spearman correlation (0.898 *vs.* 0.673). We conclude that leveraging additional features in a machine-learned model lets us overcome the aforementioned overestimation bias inherent in the source-language baseline.

To validate that highly ranked missing articles indeed attract more page views after being created, we performed a time-based evaluation by tracking readership for the set of 5.7K English articles that were missing in French Wikipedia as of June 25, 2015, but were created by July 25, 2015. Rank predictions are made based on features of the English articles before June 25, and empirical page view ranks are computed based on traffic from July 25 through August 25. Fig. 5 shows that the predicted ranks of these new articles correlate very well with their empirical ranks.

The RMSE between the predicted rank and the empirical rank is 0.173, which is higher than the offline validation step suggests (0.130; Table 2). This is to be expected, since empirical ranks were computed using page view counts over a single month directly after the article is created, whereas the model was trained on ranks computed using page view counts over six months for articles that may have existed for many years. Articles predicted to have a low rank that get created tend to have a higher empirical rank than predicted. This makes sense if the creation is prompted by news events which drive both the creation and subsequent readership, and which are not anticipated by the model.

Feature importance. Finally, to better understand which feature sets are important in our prediction task, we used forward stepwise

	Feature set added	RMSE	Spearman correlation
1	Page views	0.165	0.827
2	Topics	0.133	0.893
3	Links	0.132	0.895
4	Geo page views	0.131	0.895
5	Qual. & import. classes	0.130	0.898
6	Edit activity	0.130	0.898
7	Source-article length	0.130	0.898

Table 3: Forward stepwise feature selection results (features explained in Sec. 2.2).

feature selection. At each iteration, this method adds the feature set to the set of training features that gives the greatest gain in performance. Feature selection is done via cross-validation on the training set; the reported performance of all versions of the model is based on the held-out testing set. For an explanation of all features, see Sec. 2.2.

The results are listed in Table 3. As expected, the single strongest feature set is given by the page views the missing article gets in Wikipedias where it already exists. Using this feature set gives an RMSE of 0.165, a significant decrease from the 0.276 achieved by the source-language baseline (Table 2). Enhancing the model by adding the LDA topic vector of the source version of the missing article results in another large drop in RMSE, down to 0.133. Thereafter, adding the remaining feature sets affords but insignificant gains.

3.3 Matching editors and articles

Here we evaluate our approach to modeling editors' interests and matching editors with missing articles to be created. Recall that we model articles as topic vectors (Sec. 2.3.1), and an editor e's *interest vector* as an aggregate of the topic vectors corresponding to the articles e has edited. To measure the predictive power of these interest vectors with regard to the edits e will make next, we hold out e's most recently edited article and use the remaining most recent w articles to compute e's interest vector (where w is the *history size* parameter, *cf.* Sec. 2.3.1).

Then, all articles of the source language S are ranked by their distance to e's interest vector, and we measure the quality of the prediction as the reciprocal rank of the held-out article in the ranking. The quality of a method, then, is the the mean reciprocal rank (MRR) over a test set of editors.

Fig. 6 explores how performance depends on the history size w and the edit-history aggregation method (Sec. 2.3.1), for a set of 100,000 English Wikipedia editors who have contributed to at least two articles. We observe that the average and weighted-average methods perform equally well (considerably better than weighted medoids). Their performance increases with w up to $w = 16$ and then slowly decreases, indicating that it suffices to consider short edit-history suffixes. Under the optimal $w = 16$ we achieve an MRR of 0.0052. That is, the harmonic mean rank of the next article an editor edits is $1/0.0052 = 192$. Keeping in mind that there are 5M articles in English Wikipedia, this is a good result, indicating that editors work within topics as extracted by our LDA topic model.

4. ONLINE EXPERIMENT

The challenge we address in this research is to boost the creation of articles purveying important knowledge. We now put our solution to this challenge to the test in an *in vivo* experiment. We identify important missing articles, match them to appropriate editors based on their previous edits, and suggest to these editors by email that they might be interested in creating those articles. We focus on the source/target pair English/French and, to lower the participation threshold, give contacted editors the option to use a content translation tool built by the Wikimedia Foundation [16].

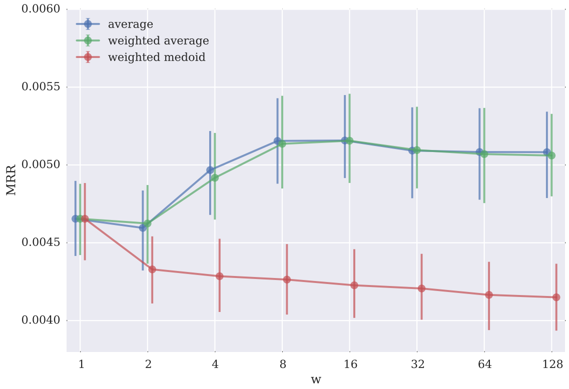


Figure 6: Mean reciprocal rank of predicting the article a user edits next, based on her previous w edits (logarithmic x-axis), with bootstrapped 95% confidence intervals; one curve per aggregation method.

To assess the effectiveness of our system, we then ask the following research questions:

RQ1 Does recommending articles for creation increase the *rate at which they are created* compared to the rate at which articles are organically created in Wikipedia?

RQ2 Do our targeted recommendations increase *editor engagement,* compared to a scenario where we ask editors to create randomly assigned important missing articles?

RQ3 How high is the *quality of articles* created in response to our targeted recommendations?

4.1 Experimental design

In order to measure the outcomes of recommending missing articles to editors, we need a set of articles as well as a set of editors. The set of articles included in the experiment consists of the top 300K English articles missing in French (Sec. 2.1) in the importance ranking (Sec. 2.2). These articles were assigned to three groups A1, A2, and A3 (each of size 100K) by repeatedly taking the top three unassigned articles from the ranking and randomly assigning each to one of the three groups. This ensures that the articles in all three groups are of the same expected importance. There were 12,040 French editors who made an edit in the last year and displayed proficiency in English (for details, *cf.* Appendix A.1.). These editors are suitable for receiving recommendations and were randomly assigned to treatment groups E1 and E2. All other French Wikipedia editors were assigned to the control group E3. Within E3, there are 98K editors who made an edit in the last year.

Based on these groupings, we define three experimental conditions (Fig. 7):

C1 **Personalized recommendation:** Editors in group E1 were sent an email recommending five articles from group A1 obtained through our interest-matching method (Sec. 2.3).

C2 **Randomized recommendation:** Editors in group E2 were sent an email recommending five articles selected from group A2 at random. (In both conditions C1 and C2, each article was assigned to at most one editor.)

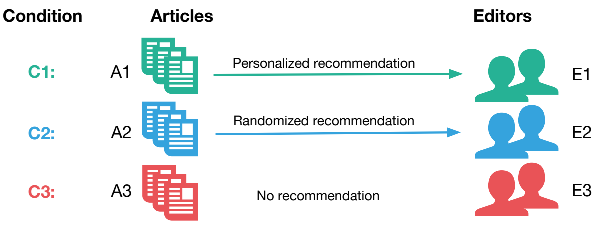
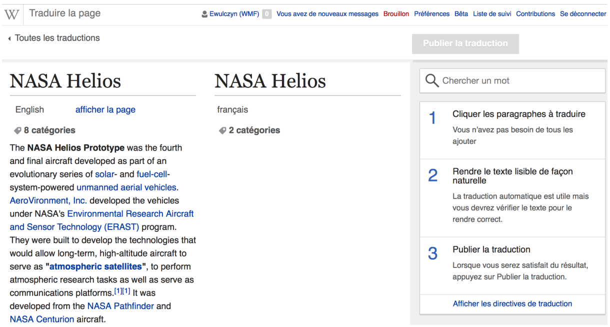


Figure 7: Experimental design of user test.



NASA Helios

English afficher la page

8 catégories

The **NASA Helios Prototype** was the fourth and final aircraft developed as part of an evolutionary series of solar- and fuel-cell-system-powered unmanned aerial vehicles. AeroVironment, Inc. developed the vehicles under NASA's Environmental Research Aircraft and Sensor Technology (ERAST) program. They were built to develop the technologies that would allow long-term, high-altitude aircraft to serve as "**atmospheric satellites**", to perform atmospheric research tasks as well as serve as communications platforms.[1][1] It was developed from the NASA Pathfinder and NASA Centurion aircraft.

NASA Helios

français

2 catégories

1 Cliquer les paragraphes à traduire

Vous n'avez pas besoin de tous les ajouter

2 Rendre le texte lisible de façon naturelle

La traduction automatique est utile mais vous devrez vérifier le texte pour le rendre correct.

3 Publier la traduction

Lorsque vous serez satisfait du résultat, appuyez sur Publier la traduction.

Afficher les directives de traduction

Figure 8: Screenshot of the content translation tool. The user is in the process of translating the NASA HELIOS article from English to French.

C3 No recommendation: Articles in group A3 were not recommended to any editor. Editors in group E3 did not receive any recommendations.

Note that not all articles from groups A1 and A2 were recommended to editors: each group contains 100K articles, but there are only about 6K editors in each of E1 and E2; since every editor received five recommendations, only about 30K of the 100K articles in each group were recommended.

Emails were sent to the editors in conditions C1 and C2 on June 25, 2015. The emails were written in French and generated from the same template across conditions (*cf.* project website [27] for exact text). To facilitate the article-creation process, each of the five recommendations was accompanied with a link to the translation tool that allows for easy section-by-section translation from the source to the target language (Fig. 8).

4.2 Results

We now evaluate the data collected through the experiment described above to answer research questions RQ1, RQ2, and RQ3.

4.2.1 *RQ1: Effect on article creation rates*

We measure the *article creation rate* for each condition as the fraction of articles in the respective group that were created in the one-month period after the recommendation email was sent. (Appendix A.2 provides methodological details on how we count the articles created in each condition.)

Comparing the article creation rates of conditions C1 and C3 will let us estimate the effect of personalized recommendation on the probability that an article is created. Further, note that the only difference between conditions C1 and C2 is that in C1 articles are assigned to editors based on our recommendation method, whereas in C2 articles are assigned to editors randomly. Therefore, by comparing the article creation rates of conditions C1 and C2, we may address the potential concern that a boost in article creation rate might be caused by the mere fact that an email recommending *something* was sent, rather than by the personalized recommendations contained in the email.

	C1	C2	C3
Potentially created	30,055	30,145	100,000
Actually created	316	177	322
Creation rate	1.05%	0.59%	0.32%

Table 4: Article creation rates for the experimental conditions defined in Sec. 4.1.

Table 4 shows the article creation rates for all experimental conditions. Important articles not recommended to any editor (C3) had a background probability of 0.32% of being organically created within the month after the experiment was launched. This probability is boosted by a factor of 3.2 (to 1.05%) for articles that were recommended to editors based on our interest-matching method (C1). On the other hand, articles that were recommended to editors on a random, rather than a personalized, basis (C2) experienced a boost of only 1.8 (for a creation rate of 0.59%).[3]

A possible confound in comparing the creation rates in C1 and C2 to C3 is the possibility that our recommendation emails diverted effort from articles in C3 and that, consequently, the creation rate in C3 is an underestimate of the organic creation rate. Although the number of editors in E1 and E2 is small (6K each) compared to the number of editors in E3 (over 98K), they differ in that editors in E1 and E2 showed proficiency in English, which might be correlated with high productivity. To address this concern, we computed the creation rate for the top 300K most important missing articles in the month prior to the experiment. We found a creation rate of 0.36%, which is only slightly higher than the rate we observed in C3 (0.32%). This indicates that the degree of underestimation in C3 is small.

A possible confound in comparing the creation rates between C1 and C2 is that, if articles matched via personalization are also predicted to be more important, then the boost in creation rate might not stem from topical personalization but from the fact that more important articles were recommended. To investigate this possibility, we compare the predicted importance (Sec. 2.2) of the articles recommended in condition C1 and C2. Fig. 9 shows that the two distributions are nearly identical, which implies that the boost in article creation rate is not mediated merely by a bias towards more popular articles among those recommended in C1.

We conclude that recommending articles to suitable editors based on their previously edited articles constitutes an effective way of increasing the creation rate of articles containing important knowledge. Although some of this increase is caused merely by reminding editors by email to contribute, the quality of the specific personalized suggestions is of crucial importance.

4.2.2 *RQ2: Effect on editor engagement*

We continue our evaluation with a more editor-centric analysis. To gauge the effectiveness of our method in terms of editor engagement, we pose the following questions: What fraction of contacted editors become active in creating a recommended article? Are editors more likely to become active in response to an email if they receive personalized, rather than random, recommendations?

[3] All pairs of creation rates are statistically significantly different ($p < 10^{-7}$ in a two-tailed two-sample t-test) with highly non-overlapping 95% confidence intervals.

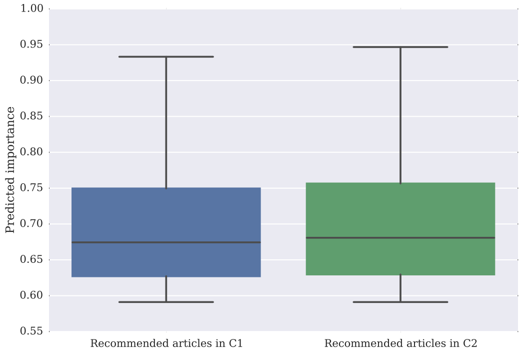


Figure 9: Box plots for predicted page view rank for articles in experimental conditions C1 and C2.

	Personal (C1)	Random (C2)
Editors contacted	6,011	6,029
Active editors	258	145
Publishing editors	137	69
Activation rate	4.3%	2.4%
Publication rate	2.3%	1.1%

Table 5: Effect of personalizing the recommendations.

Formally, we define an *active editor* as an editor who starts working on a recommended article in the translation tool (without necessarily publishing it), and a *publishing editor* as one who starts and publishes a recommended article. Given these definitions, we compute *activation and publication rates* as the fractions of all editors in each group who become active or published, respectively.

Table 5 compares these rates between the personalized (E1) and randomized (E2) editor groups (corresponding to experimental conditions C1 and C2), showing that about one in fifty editors in the randomized group (E2) started a recommended article in the translation tool (activation rate 2.4%), and that half of them went on to publish the newly created article (publication rate 1.1%). In the personalized group (E1), on the other hand, the activation (publication) rate is 4.3% (2.3%); *i.e.*, personalization boosts the activation as well as the publication rate by a factor of about two.[4] This clearly shows that personalization is effective at encouraging editors to contribute new important content to Wikipedia.

Recency of activity. The set of 12K editors who received recommendation emails included editors proficient in both English and French and having made at least one edit within the 12 months prior to the experiment (*cf.* Appendix A.1). However, being active many months ago does not necessarily imply being currently interested in editing. So, in order to obtain a more fine-grained understanding of activation rates, we bucketed editors into three roughly equally sized groups based on how many months had passed between their last edit and the experiment. Fig. 10 shows that users who were active recently are much more likely to participate, for an activation rate of 7.0% among editors with at least one edit in the month before the experiment (compare to the overall 4.3%; Table 5).

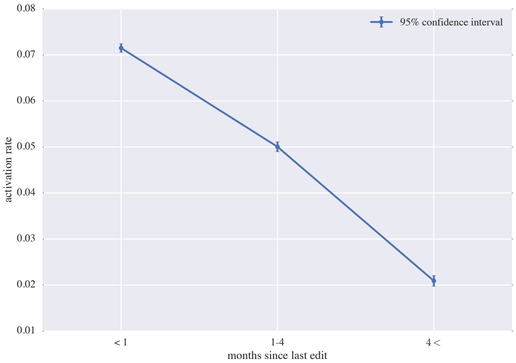


Figure 10: Editor activation rate in condition C1 as a function of months since last edit.

4.2.3 RQ3: Article quality

We conclude our analysis by evaluating the quality of articles created with the help of our recommendations. This is important as it is conceivable that editors might work more thoroughly when creating articles of their own accord, compared to our situation, where editors are extrinsically prompted to create new content.

Deletion rate. We define the deletion rate as the percent of newly created articles deleted within 3 months of being created. The deletion rate for articles in C1 that were published in response to recommendations is 4.8%, 95% CI [2.6%, 8.6%], while the deletion rate for articles in C2 that were published in response to recommendations is 9.3%, 95% CI [4.8%, 17.3%]. Note that this difference is not significant ($p = 0.063$). The aggregate deletion rate of articles published in response to recommendations (conditions C1 and C2) is 6.1%, 95% CI [3.9%, 9.4%]. In comparison, the overall deletion rate of articles created in French Wikipedia in the month following the experiment is vastly higher at 27.5%, 95% CI [26.8%, 28.2%] ($p < 0.001$).

Automatic quality score. We use the article quality classifier built for French Wikipedia [19, 12] to assess the quality of articles created by recommendation. Given an article, the model outputs the probability that the article belongs to each of the six quality classes used in French Wikipedia. Fig. 11 shows the averaged quality class probabilities for articles created and published in response to recommendations (conditions C1 and C2) and for articles that were organically created but are of similar estimated importance (condition C3) 3 months after creation. As a baseline, we also include the distribution for a random sample of French Wikipedia articles. Articles created based on recommendations are of similar estimated quality compared to articles that were organically created and the average French Wikipedia article.

Article popularity. Although not directly related to article quality, we include here a brief comparison of article popularity. Fig. 12 shows the distributions over the number of page views received in the first 3 months after creation for articles created due to a recommendation and for all other French Wikipedia articles created in the month after the start of the experiment. On average, articles created based on a recommendation attract more than twice as many page views as organically created articles.

4.2.4 Summary

In conclusion, personalized recommendations from our system constitute an effective means of accelerating the rate at which im-

[4] Publication and activation rates in the two conditions are statistically significantly different ($p < 10^{-5}$ in a two-tailed two-sample t-test) with highly non-overlapping 95% confidence intervals.

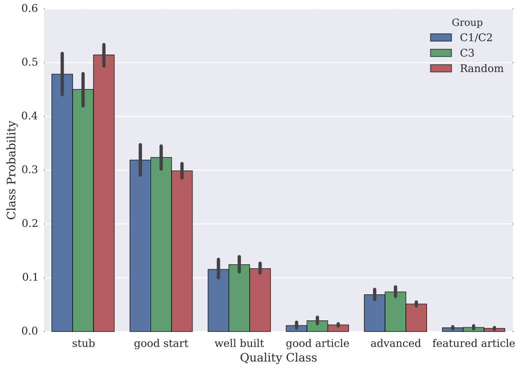


Figure 11: Aggregated quality class probabilities for articles in conditions C1 and C3, as well as for a set of 1,000 randomly selected French Wikipedia articles (with bootstrapped 95% confidence intervals).

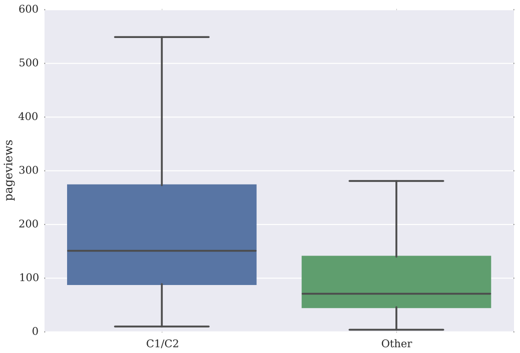


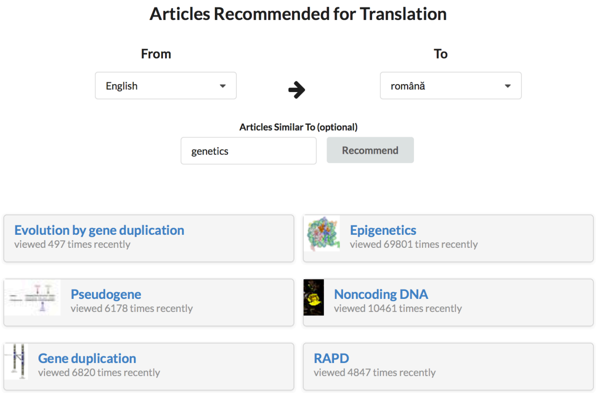
Figure 12: Box plots for the number of page views received in the first three months after creation for articles created due to a recommendation (*C1/C2*) and for all other French articles created in the month after the start of the experiment (*Other*).

portant missing articles are created in a given target language. The chance of an article recommended via personalization being created is three times that of a similar article being created organically. Further, personalizing recommendations to the emailed editor adds significant value over sending randomly selected recommendations via email, in terms of both article creation rate and editor engagement. Finally, the articles created in response to our targeted recommendations are less likely to be deleted than average articles, are viewed more frequently, and are of comparable quality.

5. DISCUSSION AND RELATED WORK

Each Wikipedia language edition, large or small, contains significant amounts of information not available in any other language [8, 13]. In other words, languages form barriers preventing knowledge already available in some editions of the free encyclopedia from being accessible to speakers of other languages [2, 28].

Until recently, very little was done to support cross-language content creation, with the exception of a few initiatives in specific topic areas such as the translation of medical information at the height of the Ebola crisis [1]. Multilingual contributors have been identified as playing a key role in transferring content across different language editions, particularly in smaller Wikipedias which still struggle to reach a critical mass of editors [11].


Articles Recommended for Translation

From: English → To: română

Articles Similar To (optional): genetics

Evolution by gene duplication
viewed 497 times recently

Epigenetics
viewed 69801 times recently

Pseudogene
viewed 6178 times recently

Noncoding DNA
viewed 10461 times recently

Gene duplication
viewed 6820 times recently

RAPD
viewed 4847 times recently

Figure 13: Screenshot of the Web application for translation recommendation.

Part of this research uses knowledge from the rich literature on personalized task recommendation. Instead of going over that literature exhaustively, we refer the interested reader to the state of the art research on personalized task recommendation systems in crowdsourcing environments [10].

In the rest of this section, we discuss how future research can help address some of the challenges we faced in our work.

Email campaigns. Email campaigns such as the one conducted in this research are limiting in several respects. Emailing recommendations involves contacting editors who may not be interested in receiving recommendations. On the other hand, editors who enjoy the recommendations may wish to receive more of them. To address these issues, we built a Web application[5] that allows users to request missing article recommendations for several language pairs. In order to make the tool useful for new editors without an edit history and to allow existing editors to explore different interests, we prompt users for a seed article. The topic vector for the seed article is used analogously to the user's *interest vector* and is used to generate personalized recommendations as described in Sec. 2.3.1. Fig. 13 shows the relevant missing articles generated by the application for a user interested in translating articles on GENETICS from English to Romanian.

Incentives to contribute. As part of this research we did not test for the impact of different incentive mechanisms. The only incentivizing information participants received was that the recommended articles were important and missing in their language. Future research on the effect of different incentive mechanisms and how much they can increase or sustain the observed boost in the article creation rate is a promising direction.

The measure of importance. In this work we rank the missing articles with respect to their predicted page views once they are created. However, it is debatable whether page views should be used as the sole measure of importance. For example, an article which is predicted to be widely read in a language may not meet the requirements for notability [26] in that language project even if the article exists in one or more other languages. This is because the notability policies and practices are sometimes different in dif-

[5] http://recommend.wmflabs.org

ferent Wikipedia language projects. Using notability as a substitute for predicted page views has the limitation that building a good training set is hard. Although many articles have been deleted from Wikipedia due to the lack of their notability, this is not the only reason for deletion and not all articles that are still in Wikipedia are notable. Research in identifying better measures of importance for article ranking can improve the quality of the recommendations.

Language imperialism and translation. An editor contacted in the experiment described recommending translations from English to French as an act of "language imperialism". Providing only English as a source language would imply that all concepts worth translating are contained in English Wikipedia, that only non-English Wikipedias need to be augmented by translation, and that out of all Wikipedia articles that cover a concept, only the English version should be propagated. A related concern is that different Wikipedia language editions cover the same concepts very differently [13] and that fully translated articles may fail to contain important information relevant to a particular language community. A major advantage of computer-supported human translation over the current state-of-the-art in machine translation is that human translators who understand the culture of the target language can alter the source where appropriate. An interesting avenue of further research would be to compare the cultural differences expressed in the initial version as well as the revisions of translated articles with their source texts.

Knowledge gaps. In this work we focused on missing articles that are available in one language but missing in another. There are multiple directions in which future research can expand this work by focusing on other types of missing content. For example, an article may exist in two languages, but one of the articles may be more complete and could be used to enhance the other. Given a method of determining such differences, our system could easily be extended to the task of recommending articles for enhancement. Alternatively, there may be information that is not available in any Wikipedia language edition, but is available on the web. The TREC KBA research track [9] has focused on this specific aspect, though their focus is not limited to Wikipedia. By personalizing the methodologies developed by TREC KBA research one could help identify and address more knowledge gaps in Wikipedia.

6. CONCLUSION

In this paper we developed an end-to-end system for reducing knowledge gaps in Wikipedia by recommending articles for creation. Our system involves identifying missing articles, ranking those articles according to their importance, and recommending important missing articles to editors based on their interests. We empirically validated our proposed system by running a large-scale controlled experiment involving 12K French Wikipedia editors. We demonstrated that personalized article recommendations are an effective way of increasing the creation rate of important articles in Wikipedia. We also showed that personalized recommendations increase editor engagement and publication rates. Compared to organically created articles, articles created in response to a recommendation display lower deletion rates, more page views and comparable quality,

In summary, our paper makes contributions to the research on increasing content coverage in Wikipedia and presents a system that leads to more engaged editors and faster growth of Wikipedia with no effect on its quality. We hope that future work will build on our results to reduce the gaps of knowledge in Wikipedia.

Acknowledgements. The authors would like to thank Dario Taraborelli for fruitful discussions, the Wikimedia Foundation Language Engineering team for technical support prior, and the French Wikipedia community for their engagement. This research has been supported in part by NSF CNS-1010921, IIS-1149837, NIH BD2K, NIH R01GM107340, ARO MURI, DARPA XDATA, DARPA SIMPLEX, Stanford Data Science Initiative, Boeing, SAP, and Volkswagen.

7. REFERENCES

[1] P. Adams and F. Fleck. Bridging the language divide in health. *Bulletin of the World Health Organization*, 93(6):356–366, 2015.

[2] E. Adar, M. Skinner, and D. S. Weld. Information arbitrage across multi-lingual Wikipedia. In *WSDM*, 2009.

[3] P. Bao, B. Hecht, S. Carton, M. Quaderi, M. Horn, and D. Gergle. Omnipedia: Bridging the Wikipedia language gap. In *CHI*, 2012.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[5] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[6] D. Cosley, D. Frankowski, L. Terveen, and J. Riedl. SuggestBot: Using intelligent task routing to help people find work in Wikipedia. In *IUI*, 2007.

[7] J. Edmonds. Maximum matching and a polyhedron with 0,1-vertices. *J. Res. Nat. Bur. Standards B*, 69:125–130, 1965.

[8] E. Filatova. Multilingual Wikipedia, summarization, and information trustworthiness. In *SIGIR Workshop on Information Access in a Multilingual World*, 2009.

[9] J. R. Frank, D. A. Max Kleiman-Weiner, N. Feng, C. Zhang, C. Ré, and S. I. Building an entity-centric stream filtering test collection for TREC 2012. In *TREC*, 2012.

[10] D. Geiger and M. Schader. Personalized task recommendation in crowdsourcing information systems: Current state of the art. *Decision Support Systems*, 65:3–16, 2014.

[11] S. A. Hale. Multilinguals and Wikipedia editing. In *WebSci*, 2014.

[12] A. Halfaker and M. Warncke-Wang. Wikiclass, 2015. `https://github.com/wiki-ai/wikiclass`.

[13] B. Hecht and D. Gergle. The tower of Babel meets Web 2.0: User-generated content and its applications in a multilingual context. In *CHI*, 2010.

[14] T. Langel. UA Parser, 2015. `https://github.com/tobie/ua-parser`.

[15] M. Manske. Not in the other language. `https://tools.wmflabs.org/not-in-the-other-language/`.

[16] MediaWiki. Content translation, 2015. `https://www.mediawiki.org/wiki/Content_translation`.

[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *J. of Mach. Learn. Res.*, 12:2825–2830, 2011.

[18] R. Řehůřek and P. Sojka. Software framework for topic modelling with large corpora. In *LREC Workshop on New Challenges for NLP Frameworks*, 2010.

[19] M. Warncke-Wang, D. Cosley, and J. Riedl. Tell me more: An actionable quality model for Wikipedia. In *WikiSym*, 2013.

[20] Wikidata, 2015. `https://www.wikidata.org/wiki/Wikidata:Main_Page`.

[21] Wikipedia. Babel template, 2015. `https://en.wikipedia.org/wiki/Wikipedia:Babel`.

[22] Wikipedia. Importance assessments, 2015. `https://en.wikipedia.org/wiki/Wikipedia:WikiProject_Wikipedia/Assessment#Importance_assessment`.

[23] Wikipedia. List of articles every wikipedia should have, 2015. `https://meta.wikimedia.org/wiki/List_of_articles_every_Wikipedia_should_have`.

[24] Wikipedia. List of Wikipedias. Website, 2015. `https://meta.wikimedia.org/wiki/List_of_Wikipedias`.

[25] Wikipedia. Quality assessments, 2015. `https://en.wikipedia.org/wiki/Wikipedia:WikiProject_Wikipedia/Assessment#Quality_assessments`.

[26] Wikipedia. Wikipedia:Notability, 2015. `https://en.wikipedia.org/wiki/Wikipedia:Notability`.

[27] E. Wulczyn, R. West, and L. Zia. Project website. `https://meta.wikimedia.org/wiki/Research:Increasing_article_coverage`.

[28] C.-M. A. Yeung, K. Duh, and M. Nagata. Providing cross-lingual editing assistance to Wikipedia editors. In *Computational Linguistics and Intelligent Text Processing*, pages 377–389. Springer, 2011.

APPENDIX

A. ONLINE EXPERIMENT: METHODOLOGICAL DETAILS

A.1 Editor and article selection

Editor selection. Only editors with high proficiency in both English and French are suitable for translating from English to French. Editors can explicitly signal their proficiency level in different languages on their user pages using the *Babel* template [21]. Editors of French Wikipedia who signaled high proficiency in English were included in the experiment.

We also included editors who made an edit in both French and English Wikipedia in the 12 months before the experiment (regardless of their use of the *Babel* template), assuming that these editors would be proficient in both languages. Since the same editor can have different user names on different Wikipedias, we use the email addresses associated with user accounts to determine which English and French accounts correspond to the same editor. We obtained a total of 12,040 editors.

Article selection. We find English articles missing in French using the method described in Sec. 2.1. We excluded disambiguation pages, very short articles (less than 1,500 bytes of content) and rarely read articles (less than 1,000 page views in the last 6 months).

A.2 Counting articles created

Counting recommended articles created. The translation tool starts logging an editor's actions after they have signed into Wikipedia, chosen a French title for the recommended missing article, and started translating their first section. This makes determining if an editor became active in response to the recommendation email easy if they used the tool. A complicating factor is that there were 39 editors who published a recommended translation (as captured by the publicly available edit logs) but did not engage with the translation tool at all. We also consider these editors to be active.

Counting articles created organically. To determine the organic creation rate of article group A3, we need to determine which newly created French articles previously existed in English, and divide by the number of articles that previously existed only in English and not in French. We observe that, within one month of being created, nearly all new French articles (92%) were linked to Wikidata, so to determine if a new article had an English version, we may simply check if the respective Wikidata concept had an English article associated with it. 62% of the new French articles meet this criterion. Further, since many newly created articles are spam and quickly deleted, we only consider articles that have persisted for at least one month. This defines the 324 articles created organically in condition C3 (Table 4).