Structuring Wikipedia Articles with Section Recommendations

Tiziano Piccardi
EPFL
tiziano.piccardi@epfl.ch

Michele Catasta
Stanford University
pirroh@cs.stanford.edu

Leila Zia
Wikimedia Foundation
leila@wikimedia.org

Robert West
EPFL
robert.west@epfl.ch

ABSTRACT

Sections are the building blocks of Wikipedia articles. They enhance readability and can be used as a structured entry point for creating and expanding articles. Structuring a new or already existing Wikipedia article with sections is a hard task for humans, especially for newcomers or less experienced editors, as it requires significant knowledge about how a well-written article looks for each possible topic. Inspired by this need, the present paper defines the problem of section recommendation for Wikipedia articles and proposes several approaches for tackling it. Our systems can help editors by recommending what sections to add to already existing or newly created Wikipedia articles. Our basic paradigm is to generate recommendations by sourcing sections from articles that are similar to the input article. We explore several ways of defining similarity for this purpose (based on topic modeling, collaborative filtering, and Wikipedia's category system). We use both automatic and human evaluation approaches for assessing the performance of our recommendation system, concluding that the category-based approach works best, achieving precision@10 of about 80% in the human evaluation.

ACM Reference Format:
Tiziano Piccardi, Michele Catasta, Leila Zia, and Robert West. 2018. Structuring Wikipedia Articles with Section Recommendations. In *SIGIR '18: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, July 8–12, 2018, Ann Arbor, MI, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3209978.3209984

1 INTRODUCTION

Wikipedia articles are organized in sections. Sections improve the readability of articles and provide a natural pathway for editors to break down the task of expanding a Wikipedia article into smaller pieces. However, knowing what sections belong to what types of articles in Wikipedia is hard, especially for newcomers and less experienced users, as it requires having an overview of the broad "landscape" of Wikipedia article types and inferring what sections are common and appropriate within each type.

Despite the importance of sections, a large fraction of Wikipedia articles does not have a satisfactory section structure yet: less than 1% of all the roughly 5 million English Wikipedia articles are considered to be of quality class "good" or better, and 37% of all articles are stubs.[1] Finally, there are many inconsistencies in section usage, even within a given Wikipedia language; *e.g.*, 80% of the section titles created in English Wikipedia are used in only one article.

Given Wikipedia's popularity and influence—with more than 500 million pageviews per day—, there is an urgent need to expand its existing articles across languages to improve their quality as well as their consistency. In other words, there is a need for a more systematic approach toward structuring Wikipedia articles by means of sections.

Fig. 1 shows the distribution of the number of sections per article for all English Wikipedia articles, alongside the same distribution for the subset of articles considered to be of high quality, according to the Objective Revision Evaluation Service (ORES),[2] a scoring system used to assess the quality of Wikipedia articles. The plot shows that over one quarter of all articles have at most one section; also, the number of sections is considerably lower when averaged over all articles (3.4), compared to the high-quality subset (7.4).

The need for developing an approach to expand Wikipedia articles is acknowledged in the literature, where the majority of the methods developed focus on automatic expansion techniques. Algorithms are developed to propagate content across languages using the information in Wikipedia's information boxes [17], to expand stubs by summarizing content from the Web [3] or from Wikipedia itself [2], and to enrich articles using knowledge bases such as DBpedia [18]. However, these approaches are limited in their

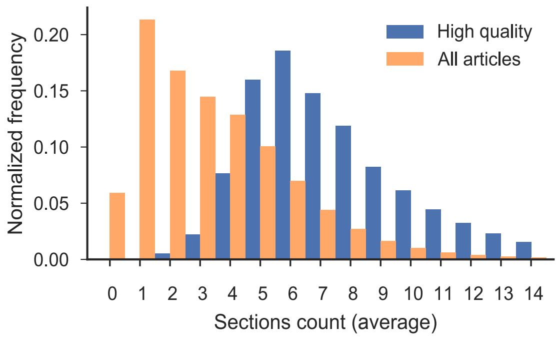


Figure 1: Distribution of number of sections per article in English Wikipedia. Good articles have more sections.

SIGIR '18, July 8–12, 2018, Ann Arbor, MI, USA
© 2018 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-5657-2/18/07.
https://doi.org/10.1145/3209978.3209984

[1]Stubs are articles considered too short to provide encyclopedic coverage of a subject.
[2]https://www.mediawiki.org/wiki/ORES

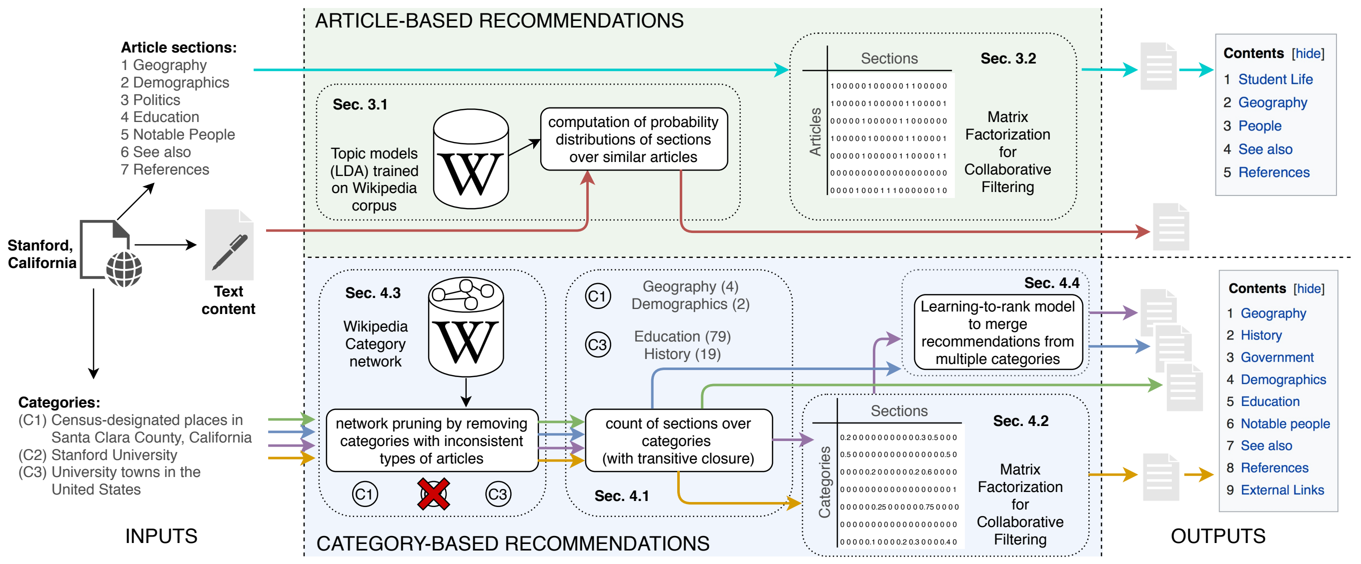

ARTICLE-BASED RECOMMENDATIONS
Article sections:
1 Geography
2 Demographics
3 Politics
4 Education
5 Notable People
6 See also
7 References
Stanford, California
Text content
Sec. 3.1
Topic models (LDA) trained on Wikipedia corpus
computation of probability distributions of sections over similar articles
Sections
Articles
Sec. 3.2
Matrix Factorization for Collaborative Filtering
Contents [hide]
1 Student Life
2 Geography
3 People
4 See also
5 References
Categories:
(C1) Census-designated places in Santa Clara County, California
(C2) Stanford University
(C3) University towns in the United States
INPUTS
Sec. 4.3
Wikipedia Category network
network pruning by removing categories with inconsistent types of articles
C1
C3
C1 Geography (4) Demographics (2)
C3 Education (79) History (19)
count of sections over categories (with transitive closure)
Sec. 4.1
Sec. 4.4
Learning-to-rank model to merge recommendations from multiple categories
Sections
Categories
Sec. 4.2
Matrix Factorization for Collaborative Filtering
CATEGORY-BASED RECOMMENDATIONS
Contents [hide]
1 Geography
2 History
3 Government
4 Demographics
5 Education
6 Notable people
7 See also
8 References
9 External Links
OUTPUTS

Figure 2: Overview of our systems for generating ranked lists of section recommendations (right) for a given input Wikipedia article (left). This paper explores several approaches, each of which is represented as a path of arrows from left to right. Different arrow colors designate different approaches. At the broadest level, we consider two paradigms: article-based recommendation (top, shaded green) and category-based recommendation (bottom, shaded blue). Each component is labeled with a reference to the section of this paper that describes it. (Best viewed in color.)

real-life applicability to Wikipedia, for several reasons. First, the type of content they generate is limited (*e.g.*, information boxes or short statements). Second, the accuracy of such approaches does not meet Wikipedia's high bar of content accuracy, which prevents such algorithms from being used in Wikipedia at scale. And third, these approaches are not editor-centric, which is in fundamental contrast to how Wikipedia is built and run.

The desirability of an editor-centric approach for empowering editors to expand Wikipedia articles, on the other hand, is acknowledged by experienced Wikipedia editors and developers through the creation of manually curated templates which are surfaced in raw format on Wikipedia howto pages as well as tools such as Ma Commune [19] and WikiDaheim [7]. These efforts aim to help newcomers to Wikipedia by providing recommendations on what sections to add to articles. The downside of these methods is that they do not scale and are very time-consuming to implement, as they rely on manually curated lists of sections per article type and for a given Wikipedia language. GapFinder [22] and SuggestBot [8] are the only automatic editor-centric recommendation systems built and used in Wikipedia. GapFinder focuses on recommending what articles to create, while SuggestBot recommends articles through calls for specific actions, such as adding citations. None of the two systems, however, addresses the need for more in-depth recommendations on *how* to expand an already existing article.

In this paper, we take thefi rst step for closing this gap in the literature by introducing and evaluating a series of editor-centric section recommendation methods. These methods differ in their source of signal (articles' topical content *vs.* Wikipedia's category network) as well as the technology used to model the recommendations (simple counting *vs.* collaborativefi ltering). We show that using Wikipedia's category network along with the proposed count-based approach provides the best recommendations, achieving precision@10 close to 80% when evaluated by human editors.

Contributions. Our main contributions are as follows.

- We introduce the problem of recommending sections for Wikipedia articles (Sec. 2).
- We devise multiple methods for addressing the problem (Sec. 3–4).
- We test our methods thoroughly in both an automatic and a human evaluation (Sec. 5–7),fi nding that methods which leverage Wikipedia's category system clearly work best.

In the evaluation section, we present results based on the English version of Wikipedia, but our method is language-independent, as it does not depend on any linguistic features.

Project repository. We make code, data, and results available at https://github.com/epfl-dlab/structuring-wikipedia-articles.

2 SYSTEM OVERVIEW

This paper addresses the problem of recommending sections for Wikipedia articles, defined as follows: given a Wikipedia article A (which may be new or pre-existent), generate a ranking of sections to be added to A.

Before describing our solutions to this problem in detail, we give a high-level overview of all the methods we explore. We support our description with the illustration in Fig. 2.

The **input** (left in Fig. 2) to all our methods consists of a Wikipedia **article** A that is to be expanded with additional sections. A is typically a stub that currently has very little content, and no, or only a few, sections. The **output** (right in Fig. 2) consists of a **ranked list** of recommended sections.

Our recommendations are intended to be screened by a human editor, who will decide which sections to incorporate into A, and who will ultimately write those sections. While there are no written rules to prevent machines from writing those sections directly in Wikipedia, in practice, such requirements are essentially imposed by Wikipedia's quality standards in a variety of its language editions.

Fig. 2 exemplifies how a section ranking is produced for the input article A = STANFORD,CALIFORNIA (the town, not the university). Note that Fig. 2 illustrates all our methods in one single diagram, each method corresponding to one colored path of arrows. At the broadest level, we explore two paradigms: article-based (top of Fig. 2; Sec. 3) and category-based recommendation (bottom of Fig. 2; Sec. 4).

Article-based recommendation (Sec. 3) works directly with article features. It leverages articles similar to the input article A to suggest sections that are contained in many of them, but not yet in A itself. Similar articles are discovered in two ways: The **topic-modeling–based** method (Sec. 3.1) leverages articles that are similar to A in terms of textual content. A mere similarity in content does not, however, necessarily make another article a good source of sections; *e.g.*, the articles on STANFORD UNIVERSITY and LELAND STANFORD have much overlapping content, but one is about a school, the other about a person, which calls for rather different article structures and thus sections. Hence, we also explore a **collaborative-filtering–based** method (Sec. 3.2), which leverages articles that are similar to A with respect to already-present sections, rather than mere textual content. (Note that this method does not apply if A has no sections yet.)

On Wikipedia, most articles are members of one or more so-called *categories; e.g.* STANFORD,CALIFORNIA , belongs to the category UNIVERSITY TOWNS IN THE UNITED STATES, among others. Our second broad paradigm, **category-based recommendation** (Sec. 4), makes use of this rich source of structured information by sourcing section recommendations for the input article A from other articles belonging to the same categories as A. In particular, for a given article A and each category C that A belongs to, our **count-based** method (Sec. 4.1) computes a score for each section S, capturing what fraction of articles in C contains S, and it then ranks sections by their scores. This method yields one section ranking for each category C that A is a member of. If A belongs to several categories, we **merge the rankings via learning-to-rank** (Sec. 4.4).

The number of possible recommendations is upper-bounded by the total number of sections contained in articles in C, which may result in a small number of recommendations for very small categories. We alleviate this problem by applying **collaborative filtering** (Sec. 4.2), which pools information between categories and allows us to make a large number of recommendations even for categories with only few member articles.

It is important to note that Wikipedia categories are organized in a network, with links representing the *subcategory* relation. Therefore, to render categories useful for our purpose, we need to reason

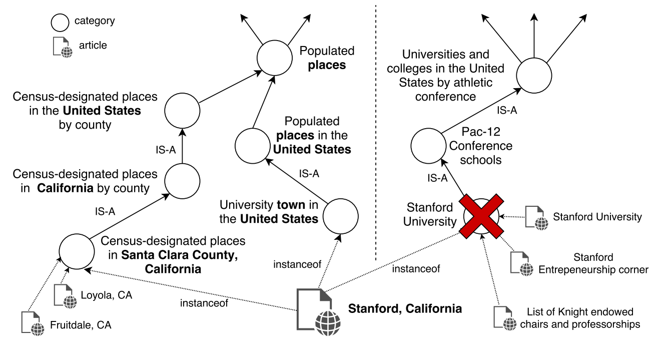

category
article
Populated places
Census-designated places in the United States by county
Populated places in the United States
Census-designated places in California by county
IS-A
IS-A
IS-A
University town in the United States
Census-designated places in Santa Clara County, California
Loyola, CA
Fruitdale, CA
instanceof
instanceof
instanceof
Stanford, California
Universities and colleges in the United States by athletic conference
IS-A
Pac-12 Conference schools
IS-A
Stanford University
Stanford University
Stanford Entrepeneurship corner
List of Knight endowed chairs and professorships

Figure 3: Category network for the STANFORD,CALIFORNIA Wikipedia article (A). In this example, the two leftmost categories (assigned to A via the *instanceof* edges) are both subcategories of the base type POPULATED PLACES. All the articles assigned to them (STANFORD,CA , FRUITDALE,CA , and LOYOLA,CA) have an *ontological* relationship with their respective categories, *i.e.*, they are instances of a populated place. We consider a category as *pure* if the majority of its assigned articles respect the ontological relationship *instanceof*. Conversely, the rightmost category STANFORD UNIVERSITY (C) is considered as impure, given the heterogeneous distribution of types of the articles assigned to C (*i.e.*, LIST OF KNIGHT ENDOWED CHAIRS AND PROFESSORSHIPS is not an instance of STANFORD UNIVERSITY). Hence, our pruning algorithm removes C from the network.

about the transitive closure of this relation, rather than about single links; *e.g.*, STANFORD,CALIFORNIA , is not explicitly labeled as a populated place; this information is only implicit in the fact that the category UNIVERSITY TOWNS IN THE UNITED STATES is connected to the category POPULATED PLACES by a sequence of subcategory links (Fig. 3). While straightforward in theory, working with the transitive closure is complicated in practice by the fact that Wikipedia's category network is noisy and ill-conceived (*cf.* Fig. 3 for an example), which gives rise to bad recommendations if not handled carefully. To avoid this problem, wefirst **prune the category network** in a preprocessing step (Sec. 4.3).

3 ARTICLE-BASED RECOMMENDATION

Intuitively, articles about similar concepts should have a similar section structure. This intuition directly translates into a strategy for recommending sections for articles: given an article A for which sections are to be recommended, detect articles similar to A and suggest sections that appear in those similar articles but that do not appear in A yet.

In order to turn this high-level strategy into a concrete implementation, we need to define what we mean by "similar articles", and we need to specify a way of transforming the set of sections contained in those articles into a ranked list of recommendations.

In this section, we introduce two variants of the article-based recommendation paradigm. Thefi rst sources its recommendations from articles that are textually similar to A, as determined by a topic modeling method (Sec. 3.1); the second, from articles that are similar to A with respect to sections that are already present, via collaborativefi ltering (Sec. 3.2).

We call these methods article-based because they operate directly on A, instead of passing through the categories of which A is a member, as done by the methods we shall introduce in Sec. 4.

3.1 Using topic modeling

Our topic-modeling approach assumes that articles containing similar topics should have a similar section structure. To this end, we build a topic model over all the articles in English Wikipedia via Latent Dirichlet Allocation (LDA) [4]. We use the LDA implementation included in the *gensim* library [15] with the number of topics set to $K = 200$.[3] In this model, an article A can be thought of as a probability distribution over the K topics.

The main step of our topic-modeling approach consists of generating a ranked list of section names per topic. To this end, we maintain K dictionaries (one per topic), whose keys are all section names. For each article A, we extract its list of sections, which is in turn used to update each dictionary: for each topic i and each section occurring in A, increment the respective entry of the i-th dictionary by A's probability for topic i. The output of this procedure is a set of K ranked lists (one per topic), where each list contains all sections names for a specific topic in order of relevance. To generate section recommendations for an article A, we build a linear combination of all rankings, weighting the ranking corresponding to topic i with A's probability for topic i, thus effectively obtaining the most popular sections from the topics that best represent A.

This method combines the advantages of content-based and popularity-based recommender models. On one hand, the content-based dimension of LDA allows us to mitigate the effect of "cold-start" articles (i.e., articles that cover novel content on Wikipedia can still be represented as a topic distribution by the topic model). On the other hand, our approach inherently recommends sections that are popular among the entire corpus, given how we build the topic-specific rankings. It is common knowledge [16] that a popularity-based recommender system can be a hard-to-beat baseline, as popular items (in our specific case, sections) are often good candidates for recommendations.

3.2 Using collaborativefiltering

The above topic modeling approach uses only the textual content of articles and ignores information about the sections that already exist in the articles. The approach we describe next does the opposite: it ignores textual content and focuses only on pre-existing sections. It aims to recommend sections from articles that are similar to the input article A with respect to the sections that already exist in A. (If A has no sections yet, this method cannot be applied.)

This is a typical collaborativefi ltering (CF) setup. Usually, CF is used to recommend items to users; given a user to make recommendations for, itfi nds similar users and suggests items those users have liked. In our setting, articles correspond to users, sections to items, and liking an item to the presence of a section.

[3]Among the different values we tested in the range from 10 to 500, setting the parameter to 200 topics provided the best trade-off between accuracy and runtime performance. The quality of the recommendations generated by our baseline approach quickly plateaued for values above 200.

We use alternating least squares [12], a standard CF method based on matrix factorization. We represent the data as a binary article–section matrix M, in which $M_{ij} = 1$ if article A_i contains section S_j, and $M_{ij} = 0$ otherwise. The matrix M is then decomposed into two factor matrices U and V such that $M \approx UV^T = \tilde{M}$ and such that U and V have a low rank of at most k. The rows of U (V) represent articles (sections) in k latent dimensions, so $\tilde{M} = UV^T$ captures the similarity of each article–section pair with respect to the latent dimensions and can thus be used to recommend new sections: *e.g.*, to suggest sections for article $A = A_i$, we sort the i-th row of $\tilde{M}$ in descending order and keep only the sections that are not already included in A.

4 CATEGORY-BASED RECOMMENDATION

Wikipedia articles are organized in a vast, user-generated network of so-called *categories*. Ideally, links in this network would represent IS-A relationships and the network would be a taxonomical network, *i.e.*, a tree-structured, hierarchical grouping of articles into a coherent ontology by subject. Unfortunately, this is not the case in practice, and before building recommendations using the category network, we need to clean it (Fig. 3). This section willfi rst explain how we can leverage the category network under the assumption that it represents a clean ontology (Sec. 4.1–4.2); then, in Sec. 4.3, we will describe how the ill-structured network can be preprocessed such that it more closely resembles a clean ontology by preserving only the likely IS-A relations in the graph. Finally, in Sec. 4.4, we will describe how the actual recommendations are generated for a given input article A by combing the relevant sections of different categories (if A belongs to several categories).

4.1 Using category–section counts

Given a category C as the entry point for generating recommendations, a simple and effective way to measure the relevance of a section S is to count the number of times S appears in all the articles in C. Concretely, we proceed as follows:

(1) As mentioned above, we assume for now that the category network provides a clean ontological structure, so we may obtain the set of all articles belonging to category C via the *transitive closure* of the subcategory relation: an article A is a member of category C if it is either a direct member of C or a member of a category in the transitive closure of C.
(2) For each section title S, we count how many of C's member articles contain a section titled S. Since category sizes—and thus section counts—may differ significantly, we normalize the counts by the number of articles in category C. The resulting scores may be interpreted as $P(S|C)$, *i.e.*, the probability of observing each section S in a randomly drawn article from category C.
(3) The ranked list of recommendations is then given by sorting the set of all sections S in decreasing order of $P(S|C)$.

This procedure requires a clean, ontological category network, which is not given off the shelf (Fig. 3). We address this issue in Sec. 4.3, where we provide a method for cleaning the category network before passing it to the above-described procedure.

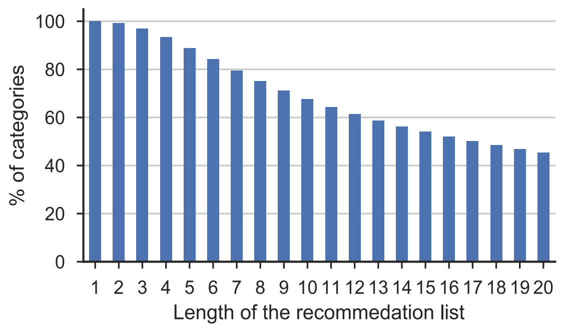


Figure 4: Percentage of categories (y-axis) that can generate at least x recommendations using the section-count–based approach of Sec. 4.1.

4.2 Generalizing via collaborativefiltering

The previous approach considers different categories as separate entities and can extract relevant sections only from the articles that are members of the respective category or of its subcategories. This is rather restrictive; *e.g.*, while SWISS SCIENTISTS and AMERICAN SCIENTISTS are semantically similar categories at the same level of the hierarchy and may thus share common sections, the counting-based approach described in the previous section cannot pool information from these two disjoint subtrees.

Therefore, we are interested in grouping similar categories to learn about missing sections by transferring information between them. This is especially important for the smaller categories, such as NAURUAN SCIENTISTS, which may contain patterns similar to other categories but do not have enough content to contribute to the generation of recommendations on their own.

To understand the impact of category size on the maximum number of recommendations we can generate, consider Fig. 4, which shows, on the y-axis, the fraction of categories large enough to generate at least x section recommendations when following the count-based approach of Sec. 4.1. As shown by the plot, *e.g.*, only 44% of the categories have enough information to produce a recommendation list of 20 sections or more.

To overcome this limitation, we leverage the fact that collaborativefi ltering can recommend sections that never occurred in a certain category, by effectively extracting signals from similar categories. We apply the same matrix decomposition method that we used for generating recommendations from articles, rather than categories (Sec. 3.2). However, in the present setup, rows of the matrix U represent categories (rather than articles) in latent space, and rows of the matrix V, sections (as before). Hence, the matrix generated by the product $\tilde{M} = UV^T$ describes the relevance of a section for a category. We tested different ways to represent the scores in the matrix, and in the evaluation section we describe the setup that gives the best results.

4.3 Cleaning the category network

As mentioned above, the category network is unfortunately rather noisy, and we cannot rely on the original network structure to extract a clean concept hierarchy. In Wikipedia, editors interpret and use the categories in different ways, and the resulting graph includes as links both subcategory relations and generic, unlabeled relations representing a variety of different ways in which two categories can be linked. For instance, the category ALGORITHMS is marked as a subcategory of APPLIED MATHEMATICS, although it is not the case that "an algorithm is an applied mathematics". Indeed, APPLIED MATHEMATICS is not even a category in the sense of "a collection of entities of the same type"; rather APPLIED MATHEMATICS is itself an entity. We refer to such "categories" as *non-ontological.* What is needed is a method for cleaning the category network by removing non-ontological categories and edges that do not encode subcategory relationships.

Prior approaches. Extracting an ontological structure from the category network is a problem that has been addressed in the past with different methods and for different purposes, but it remains an open problem. For example, MultiWibi [10] extracts a taxonomy from Wikipedia's category network for multiple Wikipedia language editions, using English as a pivot language. Unfortunately, since the source code is not public, we could not run this method on a recent snapshot of Wikipedia. When experimenting with an older snapshot, we further noticed that MultiWibi removes too many categories from the network for our purposes.

Another method [11], proposed for English Wikipedia, relies on language-specific heuristics, which poses an undesirable limitation for us, as we strive to build a language-independent system.

A third approach [5] that we tried is based on the centrality scores of nodes in the category networks. It imposes a total order on the set of all categories by ordering them in decreasing order of centrality and discarding all links that mark a lower-centrality category as the parent of a higher-centrality category. While simple and intuitively appealing, we found this heuristic to not work well for our purposes in practice; *e.g.*, it marks MONKEYS as a subcategory of CATARRHINI (a class of monkeys) because the latter is more central in the category network than the former, although the relationship should in fact be reversed.

Our approach.[4] We start from the basic insight that most of the problems of the category network are caused by non-ontological categories, *i.e.*, categories that do not represent collections of entities of the same type. As an example, consider the category (not the article) STANFORD UNIVERSITY of Fig. 3: Stanford University is itself an entity, not a collection of entities (no entity can be "a Stanford University"), yet there is a Wikipedia category with its name; this category is in fact used like a tag, for marking articles and categories that are merely somehow related to Stanford University. Treating this spurious, non-ontological category the same way we treat proper, ontological categories has a polluting effect when building the transitive closure (Sec. 4.1) of the subcategory relation, as exemplified by Fig. 3: *e.g.*, as STANFORD ENTREPRENEURSHIP CORNER is a member of the (spurious) category STANFORD UNIVERSITY, and the latter is a subcategory of PAC-12 CONFERENCE SCHOOLS, STANFORD ENTREPRENEURSHIP CORNER is erroneously classified as a member of PAC-12 CONFERENCE SCHOOLS. Such errors may pollute section counts for a vast number of categories as they percolate upward, and are amplified, in the category network.

[4]Code and data available at https://github.com/epfl-dlab/WCNPruning

Listing 1: Category-network pruning

```
def prune(current_node):
    type_hist = current_node.get_type_histogram()
    for c in current_node.children:
        child_hist = prune(c)
        type_hist = type_hist + child_hist
    if purity(type_hist) > threshold:
        mark current_node as pure
        return type_hist
    else:
        remove current_node from category network
        return an empty histogram
```

Recognizing non-ontological categories as the root of most problems in our setup, we aim at cleaning the category network by removing such categories, as well as all their incoming and outgoing edges. Note that this may diffract the category network into several disjoint connected components. This is, however, not a problem in our specific use case, as we do not require a perfect, full-coverage category network, but rather one that simply lets us compute reliable section counts for all categories.

The key operation in our algorithm is detecting non-ontological categories. In doing so, we build upon the simple intuition that a category that contains articles of more than one type (*e.g.*, persons, places, and events) cannot be a proper, ontological category; *e.g.*, the aforementioned spurious category STANFORD UNIVERSITY has (among others) persons, organizations, and events as member articles and should therefore be removed. Since the Wikipedia category network itself is not clean (which is the very reason for designing this algorithm), we of course cannot rely on it alone to determine whether a category is ontological. Instead, we must use an external resource. We use DBPedia [1], but one might also use alternative sources, such as Freebase [6] or Wikidata [20]. Concretely, we extract from DBPedia, for each article, its top-level type in the DBPedia type hierarchy (there are 55 top-level types).[5] For each category, we then construct a *type histogram*, which summarizes the DBPedia types of the articles contained in the category, and model the homogeneity, or *purity*, of the category as the Gini coefficient of its type histogram. A low Gini coefficient means that a histogram distributes its probability mass more evenly over the 55 DBPedia types, which indicates an impure, non-ontological category.

In order to prune the category network by removing impure categories, our method proceeds bottom up, starting from the sinks of the category network and propagating the set of their member articles to their respective parent categories, but only if the purity of the type histogram is above a predefined threshold. The propagated articles are then counted as part of the parent category and contribute to its type histogram.

Listing 1 shows pseudocode for a recursive implementation of this algorithm. We note, however, that, while the pseudocode captures the conceptual idea of our algorithm, we developed a more efficient implementation based on dynamic programming.

[5]The DBPedia top-level type AGENT is too general, so we break it up into its subtypes, *viz.*, PERSON, ORGANIZATION, DEITY, *etc.*

4.4 Combining recommendations from multiple categories

One article can be part of multiple categories, so when making recommendations, we should take all of them into account by merging the recommendations produced by each category, in order tofind a set of relevant sections for the article. A simple way to merge the recommendations while promoting the most representative sections is to sum their scores when they are present in more than one list. As effective as this strategy could be, it does not take into account other interesting features of the categories, *e.g.*, the number of its member articles, its type purity as captured by the Gini coefficient (Sec. 4.3), *etc.* Intuitively, such features capture the diversity of the recommendations generated by a category: the larger a category, and the lower its Gini coefficient, the more heterogeneous the recommended sections will generally be.

This problem setting is very similar to the one described in the vast body of literature on learning-to-rank (L2R) [13]. L2R was originally conceived to improve the performance of information retrieval systems, but its supervised machine learning models can also be applied to our scenario.

We develop a regression model with features based on the size of a category and its Gini coefficient. The features are generated by polynomial expansion (up to order 4), with the addition of logarithmic and exponential functions. After a round of feature selection and an optimization phase based on precision and recall (obtained on a validation set), we obtain a model capable of effectively merging the recommendations coming from multiple categories. We describe the results in Sec. 7.3.

5 EVALUATION DATASET

We test our methods using the English Wikipedia dump released on September 1, 2017, which encompasses 5.5M articles and 1.6M section occurrences. Fig. 1 shows the distribution of the number of sections per article in this dump and confirms that Wikipedia is in dire need of a strategy for supporting editors in expanding the content of articles, with 27% of articles having none or only one section. Parsing the dump, we also observed that 37% of all pages arefl agged as stubs.

Taking a closer look at the data, we notice that the most frequent sections include titles that are generic and content-independent, *e.g.*, SEE ALSO, REFERENCES, or LINKS. These sections add useful metadata to the article but do not contribute to the expansion of the content itself. If we discard a list of 14 section titles of this kind, over 54% of all articles have none or only one section.

In the dataset, we can also observe the effect of a lack of indication provided to editors by the standard editing tool: more than 1.3M section titles appear only one single time, which is frequently because editors chose titles that are too specific. For instance, the section MONUMENTS, IMAGES AND COST would be more consistent with the rest of Wikipedia if it were simply titled MONUMENTS, and instead of EXAMPLES OF OPTOMETERS, the more straightforward and more generic section EXAMPLES would have been better suited.

Dataset preprocessing. Before evaluating our methods, we clean the dataset to remove irrelevant sections that we do not want to include in our training phases. First, we remove all the unique sections, which do not represent a relevant signal for recommendation. Then, we remove a list of 14 sections (REFERENCES, EXTERNAL LINKS, SEE ALSO, NOTES, FURTHER READING,BIBLIOGRAPHY ,SOURCES , FOOTNOTES, NOTES AND REFERENCES, REFERENCES AND NOTES,EXTERNAL SOURCES,LINKS , REFERENCES AND SOURCES, EXTERNAL LINKS), which we manually selected from among the most common titles. The rationale behind this is that these sections are generic enough to be recommended by default for nearly every article, so they should not occupy spots in the short and concise, context-specific rankings we aim to provide. The evaluation results that we describe in the next section are based only on the remaining, context-specific recommendations. Moreover, all the articlesfl agged as stubs are removed to avoid training and evaluating the system on articles that have been labeled explicitly as inadequate.

After this cleaning step, 215K sections and 2.9M articles with at least one section remain. This portion of the dataset represents a core set of articles that contain enough content from which to extract common patterns. The distribution of section counts of Fig. 1 refers to thisfi ltered subset.

Dataset split. We split the entirefi ltered article set into three parts: we use 80% of the articles for training the different methods, 15% for testing, and 5% as a validation set for hyperparameter tuning.

Category network. To generate the pruned category network, we started from the category graph of the English edition released as a SQL dump in June 2017. From the original dataset composed of 1.6M nodes, we removed maintenance categories like ALL ARTICLES NEEDING ADDITIONAL REFERENCES or WIKIPEDIA PAGES REFERENCED BY THE PRESS by keeping only categories in the subtree of the category MAIN TOPIC CLASSIFICATION. This step reduces the number of categories to 1.4M. Then, using a heuristic method for breaking cycles in the graph by isolating a so-called feedback arc set [9], we removed 4,011 edges from a total of 3.9M. After this step, the category network is a directed acyclic graph.

To select the best Gini-coefficient threshold for pruning (Sec. 4.3), we use a manually annotated dataset with 710 samples. We collected this data through a Web interface that showed an article A and a category C randomly selected from the set of all of A's ancestors. For each sample, the annotators could see the complete article content and the question "Is this a C?". Feedback was provided via two buttons labeled "Yes" and "No".

This dataset contains a set of positive and negative examples of paths that should or should not, respectively, appear in the network. We pruned the graph with different Gini-coefficient thresholds and computed precision and recall using this data as the testing set.

Based on these results, we selected the break-even point of precision and recall (0.71),fi xing the Gini threshold to 0.966. This threshold value drops from the category network the most impure 5.7% of categories and removes many of the paths that would otherwise propagate the impurity to other nodes.

Performance metrics. Given an article from the testing set, our goal is to reconstruct its set of sections as completely as possible, with as few top recommendations as possible. For each article, we obtain its precision@k as the fraction of the top k recommended sections that are also contained in the testing article, and recall@k as the fraction of sections from the testing article that also appear among the top k recommendations. Taking the average of all article-specific precision@k (recall@k) values yields the global precision@k (recall@k), which we plot as a function of k, for $k = 1, \ldots, 20$ (as 20 seems a reasonable number of recommended sections to show to a user in an editing tool).

While precision and recall trivially lie between 0 and 1, not the entire range is feasible: if the number of sections in the testing article is n, then recall@k is upper-bounded by k/n, which is less than 1 if $k < n$ (*i.e.*, if we are allowed to make only few recommendations); and precision@k is upper-bounded by n/k, which is less than 1 if $n < k$ (*i.e.*, if the testing article has only few sections). When assessing our performance (Fig. 5), it is important to keep these upper bounds in mind.

6 EVALUATION: ARTICLE-BASED RECOMMENDATION

This section discusses the results obtained using the article-based recommendation methods of Sec. 3, which suggest sections based on the textual and section content of the input article. As we shall see, neither method yields results that would make it suitable for a real-world deployment scenario.

6.1 Using topic modeling

Fig. 5(a) summarizes the performance achieved by the topic-modeling approach (Sec. 3.1). Precision and recall are plotted in blue, the theoretical upper bounds (*cf.* Sec. 5) in red. The method achieves a maximum recall (at $k = 20$) of 30%. Precision is considerably lower, at 20% for the single top recommendation, and it decreases to less than 5% for $k = 20$.

Inspecting the recommended sections manually, we found that the major shortcoming of this method—rather intuitively—is that it can only capture textual, not taxonomic, similarity. A typical example of how the topic-based method may yield suboptimal results is given in Fig. 2 (top right): sections recommended for the input article STANFORD,CALIFORNIA , are effectively sourced from the article STANFORD UNIVERSITY (*e.g.*, the section STUDENT LIFE), as these two articles are very similar in terms of words contained, and therefore in terms of the topics extracted via LDA. In fact, however, the two articles call for widely different sections, one being a town, the other, a university.

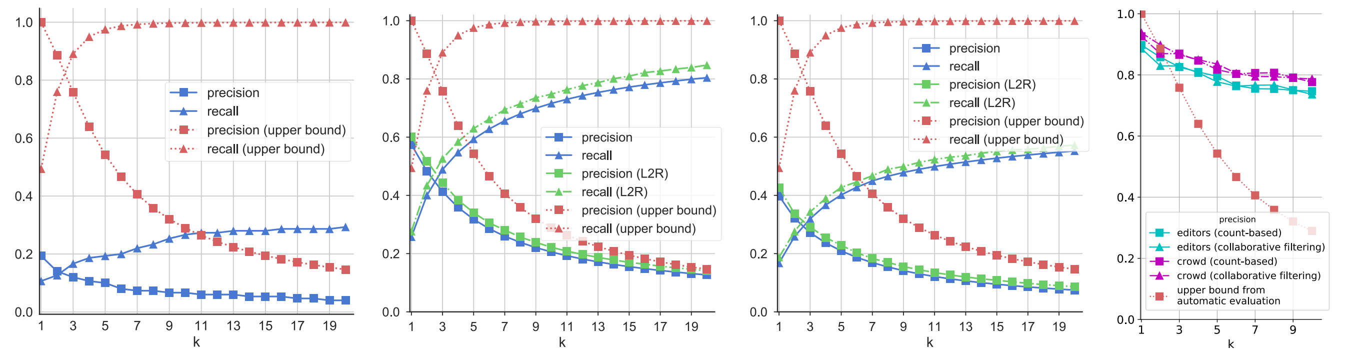
6.2 Using collaborativefiltering

The article-based collaborative-filtering method (Sec. 3.2) makes it hard to generate recommendations for articles that were not part of the training set. For this reason, we designed an evaluation setup where the goal is to reconstruct half of the sections for the articles of the testing set. In the training phase, we included all the articles of thefi ltered subset (2.9M), and we removed 50% of the sections only from the rows that are part of the testing set. We excluded the articles of the testing set with less than two sections, so we have at least one element to learn from and one to test on. This step reduced the testing set to 277K articles.

As described in Sec. 3.2, we trained the model by factoring the article-by-section "rating" matrix with alternating least squares.

Table 1: Top 5 section recommendations for the Wikipedia article LAUSANNE, according to various methods.

Topic modeling (Sec. 3.1)	**Article-based collab.filtering** (Sec. 3.2)	**Category–section counts** (Sec. 4.1)	**Generalizing counts via collab.filtering** (Sec. 4.2)
HISTORY	HISTORY OF THE DOCUMENT	HISTORY	HISTORY
SPORTS	FAMOUS RESIDENT	DEMOGRAPHICS	CAREER
AWARDS	COMMUNES WITHOUT ARMS	ECONOMY	PERSONAL LIFE
MEDAL SUMMARY	CONTENT AND IMPORTANCE	EDUCATION	HONOURS
STATISTICS	PLAYER MOVEMENT	POLITICS	CAREER STATISTICS



(a) Topic modeling (Sec. 3.1) (b) Category–section counts (Sec. 4.1) (c) Generalizing counts via CF (Sec. 4.2) (d) Human evaluation (Sec. 7.4)

Figure 5: Precision and recall as a function of the number of recommended sections k, for all methods we evaluate.

We treat the problem as an explicit feedback setup in which the article gave a favorable “rating” to its sections.

Although the experiment is intentionally designed to make the task easier (leaving half of the sections in the training set and ignoring articles with only one section), the performance of the recommendation is unsatisfactory. In particular, precision is always below 0.2%, and recall at $k = 20$ recommended titles stays below 1.5%. Although these values are significantly better than a random baseline, where precision is below 0.002% for all k, and recall for $k = 20$ is 0.008%, this method is not suitable to be used in a real-world scenario. It is well known that matrix factorization techniques perform poorly in the face of highly sparse data (a problem commonly referred to as “cold start” [12]). As such, considering that a Wikipedia article contains 3.48 sections on average, the low precision and recallfi gures for this method are not unexpected.

7 EVALUATION: CATEGORY-BASED RECOMMENDATION

We now proceed to evaluating the category-based methods of Sec. 4. As in the previous evaluation, we report the precision and recall for different values of k to show how the system behaves with different lengths of the recommendations list.[6]

7.1 Using category–section counts

Using the training set and the cleaned category network, we compute the probability $P(S|C)$ of each section S in each category C, as described in Sec. 4.1. This step generates scores for more than 191M category–section pairs, from which we extract a mapping between each category in Wikipedia and a set of sections sorted by their relevance.

Using an 80/15/5 train/test/validation split, we compute $P(S|C)$ for all (S, C) based on the training set. To make recommendations for a given article A, rather than category C, we combine the recommendations from all of A’s categories via a simple, unweighted sum of all the category-specific scores.

Fig. 5(b) shows that, with this method, precision for the unweighted sum reaches 57% for thefi rst recommended section; at $k = 20$ recommendations, recall reaches 80%. We consider this performance sufficient for deploying the method in practice.

7.2 Generalizing via collaborativefiltering

To model the relevance of a section in a category with collaborativefi ltering (CF), it is important to choose an adequate “rating” representation. We found that the best setup to training the model is to use the probabilities generated by the count-based approach while including only the top 100 sections for each category. Since we use matrix factorization to transfer information (and thus sometimes noise) between categories, the drawback is that this method is more sensitive to noise than the previous approach. Including only the top 100 sections per category helps to reduce the impact of large categories that describe very generic concepts and contain thousands of sections. After this step, we normalize rows to sum to 1 to make them comparable.

As in the case of article-based recommendations, we factorize the matrix with alternating least squares, but for this problem we modeled the CF ratings (relevance of a section) as implicit feedback.

[6]Results available at https://github.com/epfl-dlab/structuring-wikipedia-articles

Contrary to explicit feedback, where the ratings are represented as an explicit signal provided by the user, the implicit feedback is based on the presence or absence of an event (a section in this case). Besides showing better results, in this case, the ratings are not an explicit representation of the sections appearing in a certain article, but rather an implicit signal of the most relevant sections for a specific category. We selected the training parameters with the same approach described before (Sec. 6.2).

Fig. 5(c) shows that collaborativefi ltering on the category setup performs much better than the article-based approach, but, maybe unexpectedly, it does not outperform the recommendations generated by the counting method. It seems that the aforementioned propagation of noise outweighs the advantage of generalization.

7.3 Combining recommendations from multiple categories

Fig. 5(b) and Fig. 5(c) showcase also the precision–recall curves (in green) for the learning-to-rank (L2R) merging strategy described in Sec. 4.4. For both the count-based and the collaborative-filtering method, L2R provides a performance boost both in terms of precision (3%) and recall (4%). To interpret the results, we inspected the optimized feature weights and found the learned model assigns weights to the recommendation rankings that are inversely proportional to the size of a category, and directly proportional to the Gini coefficient (*i.e.*, purity) of a category (Sec. 4.3).

7.4 Evaluation by human experts

The automatic evaluation described in the previous sections has the fundamental limitation that it introduces a negative bias in the results, thus deflating our performance numbers: Many relevant sections are not yet present in the testing articles. Also, as there are no strict rules followed by the Wikipedia editors when it comes to formulating section names, a lot of section occurrences are simple syntactical and semantical variations over the same concept (*e.g.*, AWARD, AWARDS,PRIZES). The automatic evaluation is based on the exact matches of the string representations, so it fails to return a positive result for both syntactic and semantic variations. For these reasons, we also had humans evaluate our section recommendations, to assess the full performance of our models. We run our tests with two groups: experienced Wikipedia editors ($N = 11$) and crowd workers ($N = 43$). Respectively, the two groups evaluated the section recommendations for 88 and 1,024 articles. The crowd workers of the second group were recruited on CrowdFlower, a popular crowdsourcing platform, and received \$0.10 per evaluated article. To perform this evaluation, we developed a custom Web UI which shows both the recommended section and the Wikipedia article in the same browser window, allowing the evaluators to quickly iterate over the recommendations. Each article in the evaluation sample was assigned the top 10 recommendations from our best performing methods: count-based and collaborativefiltering with learning-to-rank merging (as described in Sec. 4.4). Computing recall would require the ground truth set of all relative sections for all test articles, which we do not have, so we focus only on the precision of our recommendations as perceived by the human evaluators.

As evident from Fig. 5(d), precision@k is substantially higher in the human, compared to the automatic, evaluation, from the point of view of both expert editors and crowd workers, with a precision@1 of 89% and 96%, respectively, and precision@10 of 81% and 72%. The count-based and collaborative-filtering methods are indistinguishable here, but there is a noticeable gap between the evaluations generated by the two groups. Such a difference is not surprising, as expert editors have more strict criteria when it comes to selecting sections for a certain article, while the crowd workers assessed the quality of the recommendations more intuitively based on their pertinence for the given article.

Finally, the drop in precision as k grows is a sign that evaluators did not lazily provide all-positive labels.

8 DISCUSSION AND FUTURE WORK

Wikipedia, though big, is notoriously incomplete, with fewer than 1% of articles receiving a quality label of at least "good", and 37% considered stubs [21]. The encyclopedia needs a significant amount of contributions in order to expand its existing articles and to increase their quality.

In this research, we propose a methodology to help Wikipedia editors expand an already existing article by recommending to them what sections to add to the articles they are editing. In the rest of this section, we discuss how future work can address some of the challenges faced by this research, as well as opportunities for further research.

Entry point. In this paper, we focused on one specific pathway for contributions, where users start from a Wikipedia article as an entry point. Given our above-described setup, we note, however, that we could just as well use categories as entry points; *e.g.*, the user could specify that they want to start a new article in a certain category, and our method could suggest appropriate sections. The methods of Sec. 4 can easily be adapted to this use case, as they already rely on the categories the input article belongs to.

Improving section recommendations. There are several ways in which section recommendations can be improved in practice:

- Semantically related sections are considered as independent sections in the current approach. The quality of our section recommendations could be improved by grouping these related sections (*e.g.*, WORKS and BIBLIOGRAPHY; or LIFE and EARLY LIFE) together and recommending only one of them.
- Providing more information to the editor, beyond the section title, can help the editor learn what is expected of a section. For instance, information about the average section length, the pages a section usually links to, *etc.*, can be helpful information for the editor.
- Our method currently falls short in specifying the order in which the recommended sections should be added. The order of sections in an article can, however, be important, especially for longer Wikipedia articles with many sections.
- The recommendations in this research are built using the category network information within a given Wikipedia language. With more than 160 actively edited Wikipedia languages, there is also a wealth of information in other languages that can help when expanding articles in a given

language. In the future, it will be key to consider the multilingual aspect of Wikipedia and benefit from it, especially as these recommendations should be applicable to medium-size and smaller Wikipedia languages, where the category network may not be fully developed or the amount of content is simply too limited to source useful recommendations from within the same language.
- The current count-based recommendation approach works well on frequently occurring sections. However, one of Wikipedia's strengths is the unique view of editors reflected in its content. Research on understanding how the long tail of less frequent sections can be mined to recommend infrequent but important sections will be critical in improving the recommendations.

Human-centered recommendation approach. Wikipedia is a human-centered project, where the editors' judgment, deliberation, and curiosity play key roles in how the project is shaped and content is created. This human-centric approach creates an enormous opportunity for automatic recommendations in the sense that striving for perfect precision does not need to become the center-piece of the research, while high recall will allow us to not miss out on the most important recommendations that we would miss otherwise. The same human-centered approach to content creation, however, demands interpretability of recommendations, as any edit on Wikipedia may be challenged and reasoning behind why a specific piece of content is added to the project is key in empowering the editors to use such recommendations. The count-based recommendation system developed in this research is a good starting point for allowing us to provide interpretable reasons for our recommendations.

Beyond section recommendations. Building a recommendation system for expanding a Wikipedia article will not end at recommending what sections can be added to the article. Past research has developed technology for adding hyperlinks to Wikipedia articles [14], and further research can result in more comprehensive recommendations telling editors what images, citations, external links, *etc*., to add as well.

Production system. Finally, our system will be truly useful only once it is incorporated at scale in a production system. We believe our system's performance numbers are sufficiently high for deployment, with a precision@10 of around 80% (Fig. 5(d)), so we aim to incorporate our recommender system into Wikipedia's Visual Editor in the near future.

9 CONCLUSION

In the present paper, we have introduced the task of recommending sections for Wikipedia articles. Sections are the basic building blocks of articles and are crucial for structuring content in ways that make it easy to digest for humans. We have explored several methods, some that are based on features derived immediately from the input article that is to be enriched with sections (*e.g.*, content and pre-existing sections), and others that instead generate recommendations by leveraging Wikipedia's category system. Our evaluation clearly shows that the category-based approach is superior, reaching high performance numbers in an evaluation by human raters (*e.g.*, precision@10 around 80%). We hope to deploy our system *in vivo* in the near future, in order to contribute to the growth and maintenance of one of the greatest collaborative resources created to date.

ACKNOWLEDGMENTS

We thank Diego Sáez-Trumper from the Wikimedia Foundation, as well as Jérémie Rappaz and Gael Lederrey from EPFL, for thoughtful discussions. We are also grateful to the Wikimedia editor community members who helped with the human evaluation.

REFERENCES

[1] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. DBPedia: A nucleus for a web of open data. *The Semantic Web* (2007), 722–735.
[2] Siddhartha Banerjee and Prasenjit Mitra. 2015. Filling the gaps: Improving Wikipedia stubs. In *Proc. ACM Symposium on Document Engineering*.
[3] Siddhartha Banerjee and Prasenjit Mitra. 2015. Wikikreator: Automatic authoring of Wikipedia content. *AI Matters* 2, 1 (2015), 4–6.
[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research* 3 (2003), 993–1022.
[5] Paolo Boldi and Corrado Monti. 2016. Cleansing Wikipedia categories using centrality. In *Proc. International Conference on the World Wide Web*.
[6] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proc. ACM SIGMOD International Conference on Management of Data*.
[7] The Austrian Wikimedia Community. 2017. WikiDaheim. Website. (2017). https://www.wikidaheim.at/.
[8] Dan Cosley, Dan Frankowski, Loren Terveen, and John Riedl. 2007. SuggestBot: Using intelligent task routing to help peoplefi nd work in Wikipedia. In *Proc. International Conference on Intelligent User Interfaces*.
[9] Peter Eades, Xuemin Lin, and William F Smyth. 1993. A fast and effective heuristic for the feedback arc set problem. *Inform. Process. Lett.* 47, 6 (1993), 319–323.
[10] Tiziano Flati, Daniele Vannella, Tommaso Pasini, and Roberto Navigli. 2016. MultiWiBi: The multilingual Wikipedia bitaxonomy project. *Artificial Intelligence* 241, Supplement C (2016), 66–102.
[11] Amit Gupta, Francesco Piccinno, Mikhail Kozhevnikov, Marius Pasca, and Daniele Pighin. 2016. Revisiting taxonomy induction over Wikipedia. In *Proc. International Conference on Computational Linguistics*.
[12] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (Aug. 2009), 30–37.
[13] Tie-Yan Liu. 2009. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval* 3 (2009), 225–331.
[14] Ashwin Paranjape, Robert West, Leila Zia, and Jure Leskovec. 2016. Improving website hyperlink structure using server logs. In *Proc. ACM International Conference on Web Search and Data Mining*.
[15] RadimŘeh ůřek and Petr Sojka. 2010. Software framework for topic modelling with large corpora. In *Proc. LREC Workshop on New Challenges for NLP Frameworks*.
[16] Harald Steck. 2011. Item popularity and recommendation accuracy. In *Proc. ACM Conference on Recommender Systems*.
[17] Thang Hoang Ta and Chutiporn Anutariya. 2014. A model for enriching multilingual Wikipedias using infobox and Wikidata property alignment. In *Proc. Joint International Semantic Technology Conference*.
[18] Diego Torres, Pascal Molli, Hala Skaf-Molli, and Alicia Diaz. 2012. Improving Wikipedia with DBPedia. In *Proc. International Conference on the World Wide Web*.
[19] User:0x010C and User:Ash_Crow. 2017. Ma Commune. Website. (2017). https://macommune.wikipedia.fr/.
[20] Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledge base. *Commun. ACM* 57, 10 (2014), 78–85.
[21] Wikipedia. 2017. Wikipedia:Statistics. Website. (2017). https://en.wikipedia.org/wiki/Wikipedia:Statistics.
[22] Ellery Wulczyn, Robert West, Leila Zia, and Jure Leskovec. 2016. Growing Wikipedia across languages via recommendation. In *Proc. International Conference on the World Wide Web*.