WikiCSSH: Extracting Computer Science Subject Headings from Wikipedia*

Kanyao Han[1], Pingjing Yang[1], Shubhanshu Mishra[1], and Jana Diesner[1]

School of Information Sciences,
University of Illinois at Urbana-Champaign, Champaign, IL, 61820, USA
kanyaoh2@illinois.edu;py2@illinois.edu;
mishra@shubhanshu.com;jdiesner@illinois.edu

Abstract. Domain-specific classification schemas (or subject heading vocabularies) are often used to identify, classify, and disambiguate concepts that occur in scholarly articles. In this work, we develop, apply, and evaluate a human-in-the-loop workflow that first extracts an initial category tree from crowd-sourced Wikipedia data, and then combines community detection, machine learning, and hand-crafted heuristics or rules to prune the initial tree. This work resulted in WikiCSSH; a large-scale, hierarchically-organized subject heading vocabulary for the domain of computer science (CS). Our evaluation suggests that WikiCSSH outperforms alternative CS vocabularies in terms of coverage of CS terms that occur in research articles. WikiCSSH can further distinguish between coarse-grained versus fine-grained CS concepts. The outlined workflow can serve as a template for building hierarchically-organized subject heading vocabularies for other domains that are covered in Wikipedia.

Keywords: Hierarchical Vocabulary · Wikipedia · Computer Science.

1 Introduction

A scholarly publication can be considered as a collection of concepts. Identifying these concepts allows us to build better search interfaces[1], study temporal trends in the evolution of concept usage [13,15,17], compute conceptual expertise of authors [11], and study citation patterns in scholarly data [8,12], among other practical applications. For many domains, e.g., biomedicine, mathematics, and physics, well curated, controlled, and structured vocabularies have been developed, which are commonly referred to as subject heading vocabularies (or simply, subject headings). These subject headings index relevant concepts in a domain, and organize these concepts into a hierarchical structure (e.g., concepts and sub-concepts), which facilitates coarse-grained and fine-grained knowledge

[*] This material is based upon work supported by the Korea Institute of Science and Technology Information under Grant No. C17031. We thank Kehan Li for assistance with data annotation, and anonymous reviewers for their feedback.

[1] https://www.nlm.nih.gov/bsd/pubmed.html

organization that represents the breadth and depth of a field. Examples of prominent subject headings are Medical Subject Headings (MeSH)[2], Physics subject headings (PhySH)[3], and Mathematics Subject Classification (MSC)[4].

In the domain of computer science (CS), a commonly used classification schema is the ACM Computing Classification System (ACM CCS)[5]. While this vocabulary has been curated by CS domain experts, it is being updated more slowly than the field advances, and is comparatively small-scale: The latest version of ACM CCS was released in 2012 (and its predecessor in 1998) and contains about 2,000 subject headings, while MeSH is updated once a year and contains 25,000 subject headings. Furthermore, the ACM CCS schema contains coarse-grained concepts that are helpful for identifying and categorizing relatively broad research areas of computing, but is not designed to also capture concrete, fine-grained concepts. To remedy these shortcomings, recently, the Computer Science Ontology (CSO) [19] has been introduced as an automatically constructed ontology. CSO was extracted from scholarly papers, contains about 22,000 subject headings and semantic relations between them, and can help to identify both broad and narrow research areas of computing. However, CSO fails to distinguish between core CS concepts versus concepts related to CS that emerge at the nexus of CS and other fields through interdisciplinary work. In this paper, we refer to these related concepts as ancillary CS concepts. Examples of ancillary CS concepts include "gender" and "aircraft". Another strength of CSO is that it links each concept to multiple knowledge bases, including Wikidata and Freebase. However, CSO does not yet leverage the vast amount of human effort used to organizing knowledge in Wikipedia. To address the outlined limitations, we herein report on the extraction of a large-scale, hierarchical, and semi-curated CS vocabulary that distinguishes between coarse-grained and fine-grained concepts as well as between core and ancillary CS concepts, while being grounded in knowledge provided by many people over time in the form of the Wikipedia Category Tree (WCT)[6]. We refer to our resulting vocabulary as *Wikipedia-based Computer Science subject headings (WikiCSSH)*[7] [5]. WikiCSSH was created with a mixed methods approach to extracting CS-relevant subject headings, which included breadth first search in the WCT; followed by manual filtering, community detection, embedding-based classification, and human-created rules for removing false positives. Finally, the construction of WikiCSSH benefited from the automatic association of Wikipedia pages with Wikipedia categories, which we used for the automatic expansion of WikiCSSH to include pages affiliated with Wikipedia categories into WikiCSSH.

Our project makes two main contributions. First, we provide a large hierarchical subject headings schema for CS with more than 700,000 CS concepts

[2] https://www.nlm.nih.gov/mesh/concept_structure.html
[3] https://physh.aps.org/
[4] https://mathscinet.ams.org/msc/msc2010.html
[5] https://dl.acm.org/ccs
[6] https://en.wikipedia.org/wiki/Special:CategoryTree
[7] https://github.com/uiuc-ischool-scanr/WikiCSSH

that are divided into core and ancillary concepts. Second, our work shows how to leverage the Wikipedia Category Tree for this purpose. This methodology might serve as a template for the construction of vocabularies for other domains for which information is available from Wikipedia. This paper illustrates the challenges resulting from using Wikipedia data for this specific task, shows solutions to these challenges, and implements a workflow with human-in-the-loop processes to overcome some of these hurdles.

2 Related Work

Various domains have developed their own hierarchical, domain-specific vocabularies, such as MeSH for biomedicine. MeSH is particularly useful for practical applications due to its hierarchical and non-cyclical nature. Furthermore, MeSH, along with MEDLINE, an annotated biomedical corpus, can be used to track the evolution of biomedical concepts over time and create concept profiles of authors [13, 15, 17]. The fields of mathematics and physics also have developed domain-specific vocabularies, namely, Mathematics Subject Classification (MSC) and Physics subject headings (PhySH). Finally, there exists the Wikipedia Category Tree (WCT), which covers a large number of domains and is used to classify Wikipedia articles. WCT is curated by the Wikipedia community. For CS, ACM CSS [16] and CSO [19] are the the two prominent controlled vocabularies. A comparison of various domain-specific and cross-domain controlled vocabularies is shown in table 1.

Table 1. Comparison of existing controlled vocabularies for various domains.

Name	Type	Size	Curation	Domain
MeSH	Fine grained	25K	National Library of Medicine	Biomedicine
PhySH	Fine grained	3.5K	Americal Physical Society	Physics
PACS	Subject level	9.1K	American Institute of Physics	Physics
MCS	Subject level	6.1K	Mathematical Reviews and Zentralblatt MATH	Mathematics
CCS	Subject level	2K	Association of Computer Machinery	Computer Science
WCT	Fine grained	1M+	Wikipedia contributors	Open domain

While expert-constructed vocabularies often trade off size for quality and accuracy, automatically generated vocabularies often flip this relationship. Constructing vocabularies from structured, crowd-sourced data has become another viable approach [6,7,9,20,21]. For example, prior research has leveraged Wikipedia as a comprehensive knowledge base [9, 20], e.g., for building multilingual DBpedia [7] and temporal YAGO2 [6, 21]. Since Wikipedia and the referenced related projects are not domain-specific to CS, we herein aim to leverage Wikipedia to develop a methodology for building a domain-specific, hierarchical, and non-cyclical vocabulary that distinguishes between coarse-grained and fine-grained concepts as well as between core and ancillary CS concepts.

3 Methods

3.1 Wikipedia Category Tree

The Wikipedia Category Tree (WCT) consists of 1.6M categories with 10.9M inter-category links and 217.6M category-page links. Each category in the WCT can have multiple parents as well as multiple children. Links between categories are referred as parent-child links. Each category has multiple affiliated pages. We assume that pages affiliated with a category refer to concrete concepts within that category. In other words, a category is a coarse-grained term that refers to a relatively broad research area or topic, while a page is a fine-grained term that refers to a concrete, fine-grained concept within a category. It is important to note that WCT is not necessarily a tree as it contains circular paths, e.g., *Mathematics* → *Philosophy of mathematics* → *Formalism* → *Formal sciences* → *Mathematics* → *Philosophy of mathematics*. Furthermore, since WCT is crowd-sourced and open-domain, it contains many parent-child relationships which are not relevant for our task of identifying categories relevant to CS research concepts. For example, in the parent-child chain *Computing and society* → *Social media* → *Fiction about social media*, the category *Computing and society* is relevant to CS in our context, but the category *Fiction about social media* is not. Furthermore, *Fiction about social media* leads to additional irrelevant categories (such as *Novels about social media*), and this pattern is recursive.

3.2 Building an initial CS domain-specific subtree

To construct CS specific subject headings schema, we started by extracting an initial CS subtree (ICS) as described next (see Algorithm 1). The following categories were chosen as starting points because they represent five highest-level domains relevant to CS: *computer science*, *information science*, *computer engineering*, *statistics*, and *mathematics*. These five categories constitute the first level of our initial CS subtree, and determine the overall breadth of our vocabulary. We recursively updated ICS with all children of the categories in the current ICS using a breadth first search over WCT. Redundant categories were removed during this search since we removed all categories based on exact matches of phrases that have occurred before. This resulted in an ICS with more than 1.4 million categories, which were organized in 20 levels (depth of ICS). Overall, the extraction process performed in this first step has resulted in high recall but low precision for CS-relevant categories.

3.3 Removing false positives from the ICS

Our manual inspection of this ICS revealed a few major issues.

Algorithm 1: Building WikiCSSH

```
input  : WCT, ICS ← Initial Categories, $rules$
output: WikiCSSH
$new_{cats}$ ← ICS
while $new_{cats} \neq \emptyset$ do
    categories ← Children($new_{cats}$)
    $new_{cats}$ ← categories − ICS
    ICS ← ICS ∪ $new_{cats}$
end
ICS ← Filter(ICS, $manual$)
$communities$ ← FindCommunities(ICS)
ICS ← Filter(ICS, $communities$)
$models$ ← TrainModels(ICS)
ICS ← Filter(ICS, $models$)
ICS ← Filter(ICS, $rules$)
WikiCSSH ← ExtractPages(ICS)
```

First, as described above, we identified many categories that were not related to the domain of CS and should therefore be removed from a useful CS subject headings schema. These categories often appear in lower levels of our tree, where the inclusion of even a single irrelevant category can lead to the inclusion of a large number of that category's irrelevant children. Second, while some categories were related to CS, a few of them were not useful for our intended use. These included names of CS conferences, researchers, and CS research/teaching institutes. We consider the above two issues as cases of false positives and describe our approach for removing those in Algorithm 1. It is important to note that here, false positives and irrelevant categories are meant in reference to our purpose, i.e., building a structured vocabulary of subject headings relevant for indexing research in CS, not noise or irrelevance in Wikipedia itself. We fully acknowledge that any of the instances that we did not include in WikiCSSH might very well be excellent categories for other contexts and applications.

3.3.1 Manual annotation for first three levels: The first three levels of the ICS contained a variety of broad, important sub-domains that are relevant to CS, such as *artificial intelligence* and *algorithms and data structures*. Considering that any false negatives and false positives in these levels that might be caused by automated pruning methods can lead to a lack of significant sub-domains relevant to CS or the inclusion of core research areas from other domains, respectively, we decided to manually annotate a total of 759 categories in the first three levels for relevance for our purpose, and based on that removed 259 (32%) categories from the first three levels. Even though we also removed the children of these 259 categories, there were still around 1.4 million categories remaining in the ICS.

3.3.2 Community detection: A network with an inherent community structure can be grouped into sets (communities) of nodes such that each set is densely connected within, and weakly connected across communities [3]. In our remaining ICS, categories from the same or similar domains or sub-domains were densely connected through child-parent links, such that we can assume that CS-relevant categories would be clustered together. Considering the large size of the remaining ICS (1.4 million categories), we leveraged a widely used and fast community detection algorithm, namely, the Louvain algorithm [1]. This algorithm identified a total of 288 clusters in the remaining ICS. The largest and smallest clusters contained 243,597 categories and 1 category, respectively, and the mean and median size of these 288 clusters were 5044 and 41, respectively. To identify and remove CS-irrelevant clusters, we utilized the overlap of categories in those clusters with terms in ACM CSS and CSO. We removed 261 (94.1%) clusters with a total of 0.4 million (28.6%) categories which had no overlap with ACM CCS or CSO. Our inspection of the remaining 1 million categories showed that there were still substantial numbers of false positive categories. To address this issue, we next trained a machine learning model to predict false positive categories.

3.3.3 Embedding-based classification: Our next step for reducing false positives was to use a machine learning model to automatically distinguish relevant from irrelevant categories with high accuracy. We utilized embedding based approaches, namely Elmo [18], poincare [14] and node2vec [4] embeddings, to capture the contextual information of our texts, the structured information in our subtree, and the graph information of the child-parent links in our data. While we were able to create features through embeddings, it was difficult to obtain a training set with balanced labeled responses. Since the ratio of positive to negative categories in the remaining ICS was smaller than 1%, we were not able to label enough positive instances for model training through annotating a sample from the remaining ICS. In view of this difficulty, we considered a total of 1756 shared categories between the remaining ICS and ACM CSS or CSO as positive responses. Next, we obtained negative responses by manually annotating a sample from the remaining ICS, and collecting the children of the annotated negative responses. Since we obtained tens of thousands of CS-irrelevant categories (negative responses), we randomly sampled about 1756 categories from them to create a balanced training set by combining them with positive responses. We then utilized a multi-layer perceptron (MLP) to train a model to predict whether a category is CS-relevant or not. The cross validated (k = 10) F1 score of the model was around 90%. The Elmo-based model performed best, and the addition of node2vec features improved the performance slightly (1% to 2%). Therefore, we utilized the MLP model based on the features from Elmo and node2vec to predict whether a category is relevant to CS or not. We then applied the trained model to the remaining ICS, and removed all categories labeled as CS-irrelevant from the remaining ICS. Also, if any category was classified as CS-irrelevant, we also removed its child categories. This step removed the majority of categories from the ICS. The remaining ICS only contained about 11,000 (1.1%) categories.

3.3.4 Human-created rules After inspecting the remaining ICS, we still found a substantial number of false positive categories in it. We also saw that there were more false positive categories in the bottom levels. Since manually identifying and removing individual categories is time-consuming, we developed a set of rules or heuristics to handle patterned cases of false positives that were not captured by any of the above-mentioned steps to prune the ICS. In order to find effective rules, we randomly sampled hundreds of categories from the remaining ICS, and manually annotated whether they were relevant or not. This in-depth work revealed that most false positive categories had common parent categories, and these parent categories often shared common patterns. For example, a commonly shared parent of false positive categories was *Classification system by subject*. This category did not refer to classification methods or systems in CS, but classification schemas in other domains. We also found that the suffix *by subject* in parent categories often led to the inclusion of false positive children categories into the remaining ICS as well. Another example of patterned false positives was *Microsoft software*, which is relevant to CS in general, but irrelevant for our purpose. Therefore, we removed all categories containing the suffix *by subject* and the prefix *Microsoft*. Similarly, through filtering out the false positive categories from the sample and locating their parents by tracing bottom-up parenthood links, we identified around 50 patterns, and created corresponding rules to remove them. Overall, we removed about 4000 (35%) from the remaining ICS, and obtained 7355 categories. At this point, we had used 0.45% of the categories from WCT for WikiCSSH.

3.4 Extracting fine-grained terms

Since a CS subject headings schema should also contain fine-grained concepts within each research area, we utilized all of the pages affiliated with CS-relevant categories identified through the previous steps. Based on our assumption that pages inherit the characteristic of CS-relevance from categories they are affiliated with, we extracted pages were all relevant to CS. This step refined our WikiCSSH with 761,383 pages that were affiliated with the 7355 categories in our remaining ICS.

3.5 Final WikiCSSH

The final WikiCSSH we built consists of 7K Wikipedia Categories organized as a tree, and 761K affiliated Wikipedia pages. Each category in WikiCSSH has on average 104 affiliated pages. Inter-category parent-child links capture the research field hierarchy. Category-page links capture concepts within a research field. Each category in WikiCSSH is assigned a level based on its lowest identified level in the tree. WikiCSSH contains core CS terms (including categories and pages) in levels 1-7, and ancillary CS terms in levels 8-20 (see figure 1). Core terms are highly relevant to CS research topics or concepts, while ancillary terms mainly represent interdisciplinary research topics and concepts. Core terms in WikiCSSH account for 63.5% of the terms in WikiCSSH. Users of WikiCSSH can decide which part of our vocabulary they want to use depending on their narrow or broad definition of CS.

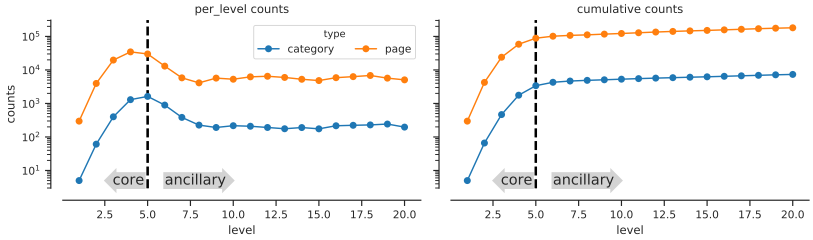

Fig. 1. Distribution of subject-heading counts in each level of WikiCSSH

4 Results and Evaluation

4.1 Comparison with other CS subject headings

Table 2 shows quantitative statistics of our final vocabulary in comparison to ACM CCS and CSO. WikiCSSH contains ∼ 7.4 thousand coarse-grained terms (categories) that are associated with ∼ 0.75 million fine-grained terms (pages) in 20 levels. Therefore, WikiCSSH is 375 and 33 times larger than ACM CCS (2,000 terms) and CSO (22,000 terms), respectively. Besides that, while both ACM CCS and CSO have a hierarchical structure to represent the relations between the terms they contain, neither of them distinguishes between coarse-grained (categories) and fine-grained (pages) terms as well as between core and ancillary terms.

Table 2. Summary of existing subject headings in Computer Science

Vocabulary	Size	Curation
ACM CCS	2K	Expert Labeling
CSO v.3.1	22K	Data Mining
WikiCSSH	7.4K categories + 752K pages	Crowdsource + HITL Data Mining

4.2 Evaluation of category extraction based on human annotated data

In this section, we evaluate the performance of our methods for removing false positives. This evaluation also allows us to test whether our mixed methods approach can outperform any single method approach to pruning a large-scale dataset with a complex structure such as the WCT. We randomly selected a sample of categories from the ICS before our community detection step, and manually annotated whether the sampled categories were CS-relevant or not. Finally, we leveraged this annotated sample to evaluate precision and recall of different category sets extracted through different methods. It is important to note that we only evaluated categories. Pages inherit the characteristic of relevance to CS from categories they are affiliated with and thus are assumed to share similar results with the evaluation for categories. Table 3 shows the evaluation results. The first three levels are not useful for evaluation as they have been selected manually. From level 4 onward, we find that the embedding based method (ML) achieved a higher precision compared to the community detection (CD) method at the expense of lower recall. Combining ML with rules also increases precision at the expense of lower recall, while combining all of CD, ML, and Rule improves the precision significantly in lower levels (more than 0.4 points for levels 6 and 7). This result provides empirical evidence for our argument that mixed methods can outperform a single method approach or pruning large-scale data with complicated structures.

Table 3. Precision (P) and recall (R) in core levels (recall for levels > 5 cannot be computed as that would require manually annotating all CS-relevant categories.)

	CD		**ML**		**ML+Rule**		**CD+ML+Rule**	
Level	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**
1-3	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
4	0.36	1.00	0.64	0.88	0.85	0.90	0.85	0.88
5	0.17	1.00	0.66	0.90	0.87	0.79	0.87	0.79
6	0.07	/	0.25	/	0.47	/	0.83	/
7	0.01	/	0.23	/	0.33	/	0.82	/

4.3 Evaluation of WikiCSSH against an annotated scholarly dataset

A common application of a subject headings vocabulary is to tag scholarly papers with these subject headings. A domain-specific subject headings vocabulary can be considered effective or to provide high coverage if it enables the identification of important key-phrases in a dataset of scholarly papers from that domain. We used KP20k dataset [2,10] for our evaluation. KP20k contains 20,000 CS research abstracts and human-annotated short key phrases from these abstract, such as *machine learning*, *data mining*, and *clustering*, among others. We matched each keyphrase in KP20k against the terms in WikiCSSH, ACM CSS, and CSO. We basically searched for all exact matches of stemmed words in the keyphrases with stemmed terms from the subject headings vocabularies. For our evaluation, we counted both the number of unique matched phrases and the total number of matched phrases. since the total number of phrase matches is going to be biased towards frequently occurred concepts that are likely to be present in all vocabularies, we also used unique phrase matches to identify coverage. As reported in table 4, WikiCSSH extracted 62,635 (8.23%) unique phrases and 1,456,690 (48.3%) total phrases from KP20k, and most of them were contributed by WikiCSSH's core part. The numbers of extracted unique and total phrases for ACM were 1,284 (0.17%) and 302,326 (10%), and for CSO 10,985 (1.44%) and 797,447

(26.4%). This evaluation suggests that WikiCSSH supports comparatively high coverage of CS terms occurring in scholarly texts.

Table 4. Comparison of coverage of various vocabularies on phrases in KP20k corpus (percents = phrases extracted by vocabulary / annotated phrases in KP20K)

Vocabulary	Unique Phrases	Total Phrases
ACM CCS	1,284 (0.17%)	302,326 (10%)
CSO	10,985 (1.44%)	797,477 (26.4%)
WikiCSSH (core)	45,345 (5.96%)	1,207,075 (40%)
WikiCSSH (ancillary)	17,290 (2.27%)	249,515 (8.27%)
WikiCSSH (total)	62,635 (8.23%)	1,456,590 (48.3%)

We also calculated the ratios of total to unique phrases, respectively, for the core and ancillary part of the WikiCSSH, which show WikiCSSH's ability to extract rare phrases from KP20K. We found that the ratio of total to unique phrases for the core part of WikiCSSH is 26.62, while for the ancillary part, it is only 14.43. Put differently, the core part of WikiCSSH is more likely to capture frequently occurring phrases in CS research articles, while the ancillary part tends to capture rare phrases. Similarly, for ACM CCS and CSO, the ratio of total to unique phrases were 235.5 and 72.6, respectively. This result indicates that WikiCSSH is more likely to extract rare phrases from scholarly articles compared to ACM CCS and CSO. CSO, which was constructed from mining large-scale scholarly data in CS, contains a lower proportion of rare phrases that occur in KP20K compared to WikiCSSH. A possible reason for this lower coverage may be automated data mining methods inability to capture low probability signals.

5 Conclusion, Discussion and Limitations

We have presented WikiCSSH, a large-scale subject headings vocabulary for the CS domain, that we developed using a human-in-the-loop workflow that leverages the crowd-sourced Wikipedia Category Tree. WikiCSSH outperforms two alternative CS vocabularies, namely ACM CCS and CSO, in number of items, coverage of key-phrases in a benchmark dataset of scholarly papers from CS, and categorization of the subject headings into coarse-grained versus fine-grained entries. Users of WikiCSSH can decide which part of WikiCSSH they want to use depending on their needs. For example, users may want to i) use the 7,355 hierarchically structured categories for indexing (research areas and topics in) documents, or ii) use the 0.75 million concrete, fine-grained terms (from pages) within categories for more detailed concept analysis, or iii) select the core and/or ancillary part of WikiCSSH according to their broad or narrow definition of computer science as needed for their work.

Our work also contributes to methodological work for leveraging existing crowd-sourced data when the main challenge is filtering out false positives to increase precision of some target application. Building a sizeable domain-specific vocabulary like WikiCSSH would be extremely expensive and/ or time consuming if one only relied on manual work by domain experts. However, existing crowd-sourced data with a permissible license opens up an opportunity to build a large-scale, structured vocabulary at low cost in terms of both time and human resources. That being said, our approach is more costly and time-consuming than a fully automated data mining- based approach due to the substantial human interventions we made part of our process. However, we showed that our approach can capture relevant yet rare phrases that might be ignored by fully automated data mining solutions. Our work also illustrates the challenges resulting from using the given structure of Wikipedia data for our specific task and assesses possible solutions to overcome these challenges through the methods described earlier. Our workflow can be extended to construct subject headings for other domains by modifying the rules and training approaches. Code for replicating the construction and refinement of WikiCSSH along with the latest version of WikiCSSH can be found at: https://github.com/uiuc-ischool-scanr/WikiCSSH [5].

We acknowledge the limitations of our evaluation of WikiCSSH, for which we aimed to map key-phrases in scholarly papers to entries in WikiCSSH. Even though WikiCSSH outperformed other domain-specific vocabularies in terms of coverage of KP20K, this result only highlights its potential to extract more CS-relevant phrases from scholarly articles than alternative vocabularies. However, precision may be more important if we aim to categorize or index documents based on a controlled vocabulary. In our future work, we plan to test the performance of WikiCSSH for analyzing scholarly data, indexing and categorizing documents, and mining phrases and topics. Additionally, because Wikipedia data and classification methods are being updated over time, we plan to update WikiCSSH based on new data and with new methods as well.

References

1. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment **2008**(10), P10008 (oct 2008). https://doi.org/10.1088/1742-5468/2008/10/p10008
2. Gallina, Y., Boudin, F., Daille, B.: Large-scale evaluation of keyphrase extraction models. arXiv preprint arXiv:2003.04628 (2020)
3. Girvan, M., Newman, M.E.: Community structure in social and biological networks. Proceedings of the national academy of sciences **99**(12), 7821–7826 (2002)
4. Grover, A., Leskovec, J.: Node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 855–864. KDD '16, Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/2939672.2939754
5. Han, K., Yang, P., Mishra, S., Diesner, J.: Wikicssh - computer science subject headings from wikipedia (2020). https://doi.org/10.13012/B2IDB-0424970_V1

6. Hoffart, J., Suchanek, F.M., Berberich, K., Weikum, G.: Yago2: A spatially and temporally enhanced knowledge base from wikipedia. Artificial Intelligence **194**, 28 – 61 (2013). https://doi.org/https://doi.org/10.1016/j.artint.2012.06.001
7. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., Van Kleef, P., Auer, S., et al.: Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. Semantic Web **6**(2), 167–195 (2015)
8. Levine, T.R.: Rankings and trends in citation patterns of communication journals. Communication Education **59**(1), 41–51 (2010)
9. Medelyan, O., Witten, I.H., Milne, D.: Topic indexing with wikipedia. In: Proceedings of the AAAI WikiAI workshop. vol. 1, pp. 19–24 (2008)
10. Meng, R., Zhao, S., Han, S., He, D., Brusilovsky, P., Chi, Y.: Deep keyphrase generation. arXiv preprint arXiv:1704.06879 (2017)
11. Mishra, S., Fegley, B.D., Diesner, J., Torvik, V.I.: Expertise as an aspect of author contributions. In: Workshop on Informetric and Scientometric Research (SIG/MET). Vancouver (2018)
12. Mishra, S., Fegley, B.D., Diesner, J., Torvik, V.I.: Self-citation is the hallmark of productive authors, of any gender. PLOS ONE **13**(9), e0195773 (sep 2018). https://doi.org/10.1371/journal.pone.0195773
13. Mishra, S., Torvik, V.I.: Quantifying Conceptual Novelty in the Biomedical Literature. D-Lib magazine : the magazine of the Digital Library Forum **22**(9-10) (sep 2016). https://doi.org/10.1045/september2016-mishra
14. Nickel, M., Kiela, D.: Poincaré Embeddings for Learning Hierarchical Representations. In: Advances in Neural Information Processing Systems 30. pp. 6338–6347. Curran Associates, Inc. (2017)
15. Nielsen, F.Å., Mietchen, D., Willighagen, E.: Scholia, scientometrics and wikidata. In: The Semantic Web: ESWC 2017 Satellite Events. pp. 237–259. Springer International Publishing, Cham (2017)
16. Osborne, F., Motta, E.: Klink-2: integrating multiple web sources to generate semantic topic networks. In: International Semantic Web Conference. pp. 408–424. Springer (2015)
17. Packalen, M., Bhattacharya, J.: Age and the trying out of new ideas. Journal of Human Capital **13**(2), 341–373 (2019). https://doi.org/10.1086/703160
18. Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep Contextualized Word Representations. In: Proceedings of the 2018 Conference of the NAACL-HLT. pp. 2227–2237. Association for Computational Linguistics, Stroudsburg, PA, USA (jun 2018). https://doi.org/10.18653/v1/N18-1202
19. Salatino, A.A., Thanapalasingam, T., Mannocci, A., Osborne, F., Motta, E.: The computer science ontology: a large-scale taxonomy of research areas. In: International Semantic Web Conference. pp. 187–205 (2018)
20. Shang, J., Liu, J., Jiang, M., Ren, X., Voss, C.R., Han, J.: Automated phrase mining from massive text corpora. IEEE Transactions on Knowledge and Data Engineering **30**(10), 1825–1837 (2018)
21. Wang, Y., Zhu, M., Qu, L., Spaniol, M., Weikum, G.: Timely yago: harvesting, querying, and visualizing temporal knowledge from wikipedia. In: Proceedings of the 13th International Conference on Extending Database Technology. pp. 697–700 (2010)