Finding Prerequisite Relations using the Wikipedia Clickstream

Mohsen Sayyadiharikandeh
Indiana University, Bloomington
Bloomington, Indiana, USA
msayyadi@indiana.edu

Jonathan Gordon
Vassar College
Poughkeepsie, New York, USA
jgordon@vassar.edu

José Luis Ambite
USC Information Sciences Institute
Marina del Rey, California, USA
ambite@isi.edu

Kristina Lerman
USC Information Sciences Institute
Marina del Rey, California, USA
lerman@isi.edu

ABSTRACT

The increased availability of online learning resources in the form of courses, videos, and tutorials has created new opportunities for independent learners, but it has also increased the difficulty of planning a course of study. Where should the learner start? What should the learner know before tackling a new course? Manually identifying these prerequisite relations between learning resources or concepts is expensive in terms of time and expertise, and it is particularly difficult to do so for new or rapidly changing areas of knowledge. To address this challenge, we present a new method for identifying prerequisite relations based on naturally occurring data, namely the navigation patterns of users on the Wikipedia online encyclopedia. Our supervised learning approach shows that the navigation network structure can be used to identify dependencies among concepts in several domains.

KEYWORDS

Prerequisite; Clickstream; Wikipedia; Concept graph

ACM Reference Format:
Mohsen Sayyadiharikandeh, Jonathan Gordon, José Luis Ambite, and Kristina Lerman. 2019. Finding Prerequisite Relations using the Wikipedia Clickstream. In *Companion Proceedings of the 2019 World Wide Web Conference (WWW '19 Companion), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3308560.3316753

1 INTRODUCTION

Self-directed learners can benefit greatly from the scientific and technical training resources available online. However, it can be challenging to organize these resources into a suitable educational plan. In traditional education, whether in the classroom or in textbooks, concepts are taught in a sequence determined by an expert's understanding of the domain. Self-directed online learners, however, may not know where to begin. For example, if your goal is to understand recursive neural networks, you may not know that you first need to understand several more basic concepts in mathematics and machine learning.

In this task, learners can be guided by a *concept graph*, which links particular educational resources – such as online courses, videos, and tutorials – to the concepts they cover and links each concept to any more basic, prerequisite concepts. We consider a concept A to be a prerequisite of concept B if A is necessary or significantly helpful for understanding B. (Other relations, such as similarity or hyponymy, can hold between concepts without either being a prerequisite of the other.) A concept graph can be used directly by a learner to explore the conceptual structure of the domain, like looking at a map, or it can be used by applications to recommend particular learning resources based on the prerequisites between concepts, as in the generation of structured reading lists [5].

Manually constructing a concept graph is time-consuming and requires significant domain knowledge, motivating automatic methods. We introduce an approach that infers prerequisite relations between concepts based on the navigation of users on Wikipedia. We treat encyclopedia articles as identifying concepts – with varying levels of granularity – and train a classifier to predict whether concept A is a prerequisite of concept B based on features computed from a graph of "clickstream" data, where articles are connected by weighted edges that indicate the number of times users followed links from one article to the other. Intuitively, users visiting an article are interested in learning about that concept, and they will follow links to other articles they believe will support that objective. Thus, navigation tends to flow from a concept to its prerequisites.

For training and testing, we use two sets of gold-standard prerequisite data. One is an existing set, which covers several domains and uses crowdsourced judgments of whether a prerequisite relation holds between a pair of Wikipedia articles [15]. The other is derived from a large, expert-generated concept graph for machine learning [7], which we semi-automatically map to the most relevant Wikipedia articles.

The contributions of this paper are: (1) a novel approach to prerequisite identification based on the observed behavior of online learners and (2) a new evaluation set for prerequisites based on a semi-automatic mapping of an expert-generated concept graph to Wikipedia.

2 RELATED WORK

Manually curated graphs of prerequisite relations are used to guide learners[1] and plan curricula – i.e., to order learning resources based

[1] For a recent example, see Figure 1.6 in Goodfellow et al.'s 2016 textbook on deep learning [4].

WWW '19 Companion, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6675-5/19/05.
https://doi.org/10.1145/3308560.3316753

on the concepts they cover. For instance, Stanford's course CS 234 *Reinforcement Learning* has as a prerequisite CS 229 *Machine Learning*, which lists linear algebra, probability, and statistics as prerequisites.[2] Several researchers have recently presented approaches to automatically infer these relations between concepts and pedagogical resources. Before presenting our approach, we briefly review these efforts.

Talukdar and Cohen [15] crowdsourced judgments of prerequisite relations, and then employed a maximum-entropy classifier with three types of features: (1) derived from Wikipedia's link graph, (2) related to Wikipedia users' edits, and (3) related to article content. Their classifier achieved 58.82% accuracy. Liang et al. [10] proposed a link-based metric for measuring prerequisite relations among Wikipedia concepts. They compute prerequisites based on reference distance (RefD), where Wikipedia links serve as "reference relations" among concepts. (Reference distance is asymmetric, so A and B cannot both be prerequisites of each other.) They evaluate on Talukdar and Cohen's prerequisite data and on their own gold-standard dataset made by crawling university websites and mapping to Wikipedia concepts. Since this new evaluation set finds prerequisites at the level of entire courses, they are significantly more coarse-grained than the Metacademy-based data set we introduce (e.g., a concept would be *Machine learning*, not *Markov chain Monte Carlo*).

Liang et al. [10] re-implemented Talukdar and Cohen's maximum-entropy method, for which they report a higher average accuracy of 60.4%, which may be due to the use of newer Wikipedia data. They also introduce a simpler method which achieves a higher average accuracy of 61.2%. In later work, Liang et al. [11] studied the applicability of an active learning approach applied to their previous method with some novel features. They employed different query strategies for pool-based active learning and concluded that query-by-committee constantly outperforms other methods. They also achieved higher accuracy compared to other related work.

Wang et al. [16] use the order of topics in textbook tables of contents to extract concept maps by jointly optimizing the extraction of key concepts with corresponding Wikipedia articles and the identification of prerequisite relations. They evaluate their method by building concept maps from six textbooks and having domain experts evaluate the results. A limitation of this approach is that it requires concepts to have already been ordered by an expert in a textbook. For many domains, appropriate textbooks may not be readily available, or a concept graph might require combining the orderings from multiple textbooks.

Medio et al. [13] considered predicting prerequisite relations between "learning objects" on Coursera. For each learning object, they find a set of related Wikipedia articles and, using Coursera's gold-standard prerequisites between courses, they train a classifier using textual and hyperlink features of the matched Wikipedia articles. Given that many learning resources can describe the same concepts, we are interested in inferring prerequisite relations among concepts and using these to form learning plans rather than predicting prerequisite relations among particular resources.

[2] http://exploredegrees.stanford.edu/schoolofengineering/computerscience

Gordon et al. [6] present a novel model where each concept – a latent Dirichlet allocation (LDA) topic learned from a corpus of scientific articles – is linked by prerequisite relations as a step in building a concept graph to support the automatic generation of personalized reading lists. Using this formulation of concepts as probability distributions over words, they introduced an information-theoretic view of prerequisite relations based on cross entropy and information flow. An advantage of this work is that it does not rely on an external reference source like Wikipedia, but a limitation is that the results of unsupervised topic discovery are harder for humans to interpret than a Wikipedia article.

2.1 Our Approach

In this paper, we use Wikipedia articles as concepts, which can be linked to learning resources, such as online courses, videos, or documents, e.g., by using Explicit Semantic Analysis [3]. While there are limitations to the use of Wikipedia articles as concepts, such as the problem of identifying sub-articles describing a facet of a more general concept [12], each article has a clear interpretation, and the set of articles has broad coverage.

Rather than use the textual or hyperlink network features of previous work, we use a new data source: the Wikipedia Clickstream, consisting of actual navigation of learners among articles. While we do not assert that this navigation replaces the information that can be learned from other features, we investigate how much it can contribute toward the problem of prerequisite identification.

Like previous work in this area, we use the Talukdar and Cohen [15] Wikipedia evaluation set. However, like Liang et al. [10], we see the need for an additional, naturally occurring evaluation set – albeit with more specific concepts than they used – leading us to create mappings from Metacademy to Wikipedia. These data sets are described in more detail in the next section.

3 DATA SETS

We use three sources of data: the Wikipedia Clickstream and two sets of prerequisite relations used for training and testing. The first is Talukdar and Cohen's multi-domain crowdsourced judgments, used by most previous work, and the second is a new data set based on Metacademy's expert-enumerated prerequisite relations, which were created to guide learners rather than to evaluate research.

3.1 Wikipedia Clickstream

The Wikipedia Clickstream [17] consists of data sets containing counts of (referrer, resource) pairs extracted from the user request logs of the English, Farsi, and Arabic editions of Wikipedia. The data is divided into months of clicks, starting with January 2015, and includes only pairs of articles with more than 10 clicks. The clickstream data can be seen as giving a weighted network of articles [9], where the weights are human navigation through a popular network of learning resources. For research projects, it is a large network; it includes 1.3 million nodes and 22 million edges for January 2017 alone. In addition to navigation between articles, it includes inbound clicks from web sites such as google.com.

We compute our features using the English Wikipedia clickstream data sets that were released for January and February 2015, six months in 2016, and January 2017. We filtered these data sets to

only those links where both articles are in the set of concepts under consideration (articles predicted to be relevant by the automatic mapping of Metacademy concepts or those articles chosen for inclusion in the CMU prerequisite data set, described in the following sections). Note that pairs retrieved at this step include not only positive examples of prerequisite pairs but also (probable) negative examples, where both concepts are of interest in the domain but are not known to have a prerequisite relation.

3.2 CMU Prerequisite Data

The CMU prerequisite data [15] covers five domains: Global Warming, Meiosis, Newton's Laws of Motion, Parallel Postulate, and Public-key Cryptography. For each domain, Talukdar and Cohen selected pairs of articles based on their random walk with restart (RWR) scores. At the time the data was collected, every pair of articles (d, d') in their data set had a hyperlink from d to d'. These pairs were then presented to people on Amazon Mechanical Turk, who were asked whether (1) d' is a prerequisite of d, (2) d is a prerequisite of d', (3) the articles are unrelated, (4) the articles are related but there is no prerequisite relation, or (5) don't know. For the Newton's Laws and Global Warming domains, they collected five votes for each of 400 pairs per domain. For the Parallel Postulate and Public-key Cryptography domains, they collected three votes for each of 200 pairs per domain. For Meiosis, they collected three votes for each 400 pairs. A limitation of these data sets is their modest scale, including fewer than 100 positive pairs of prerequisites per domain.

While naturally occurring sources of prerequisites – like Metacademy, described next – consist of only positive examples, the CMU data set includes three types of negative samples: unrelated pairs, pairs that are related but not prerequisites, and when a prerequisite was identified as being in the other direction. Following the processing described by Talukdar and Cohen, we excluded "Don't know" responses, aggregated the votes, and assigned final labels based on majority vote for each pair, breaking ties arbitrarily. We considered the three mentioned types of relations between pairs as negative examples for the final data sets.

3.3 Metacademy Prerequisite Data

Metacademy (metacademy.org) is a free, open-source platform for learning. It covers 487 concepts related to machine learning, which are connected in 1,208 prerequisite pairs (7,947 under transitive closure). Figure 1 shows a sample Metacademy learning plan, where each edge represents a prerequisite relation. To use the expert-generated Metacademy prerequisite relations for the problem of inferring prerequisites among Wikipedia concepts, we introduce a method to map Metacademy concepts to the closest matches in Wikipedia's semantic space.

Finding Possible Wikipedia Matches. Our initial set of possible matches comes from querying a search engine, since this lets us exploit significant relevance engineering, including features based on human intelligence (since search engines generally take human clicks into account when ranking results). In particular, we searched DuckDuckGo for "*Metacademy concept* + wiki page" and excluded results with URLs that were not from Wikipedia.[3] The granularity of concepts on Metacademy sometimes differs from that of articles on Wikipedia, so some concepts in learning plans do not have exactly corresponding Wikipedia articles, either because they are too specific (e.g., *Higher-order partial derivatives* or *Linear regression closed-form solution*) or because they are too general (e.g., *Expectation and variance*).

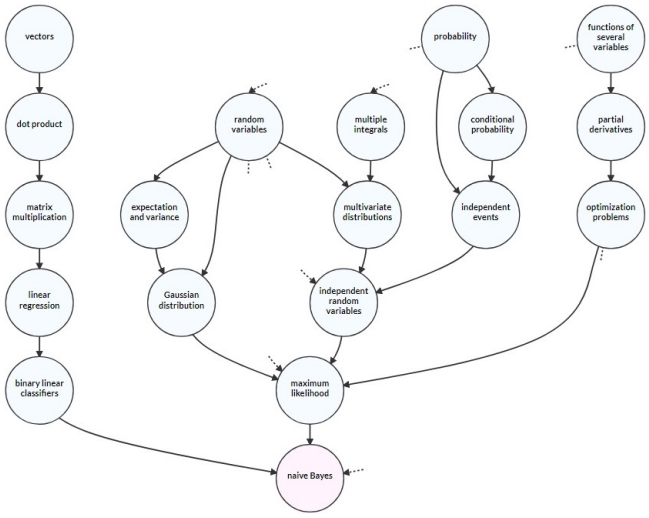

Figure 1: Sample learning plan from Metacademy

Predicting Relevance of Matches. For a sample of 100 Metacademy concepts, 763 search results were annotated as 0 (not related), 0.5 (related), or 1 (exact match, either for the whole article or a section of it). The annotation was performed by two of the authors, with a Pearson correlation of 0.75 ($p < 0.001$). We found that about 89% of samples that were labeled as 1 by both annotators also have rank 1 in search results. So, a simple baseline classifier predicts that the first result for every query is a correct match.[4] To distinguish which of the additional results are likely to be related (0.5) vs not (0), we use the "Category" of the Wikipedia articles: An article whose set of categories overlaps with the categories of the rank-1 result for that query will be labeled as 0.5. If there is no overlap, then the article is likely to be a spurious match. Based on manual inspection, we included two levels of ancestor categories when determining overlap.

As an example, if we search for "d separation", three Wikipedia articles are found in the first page of results: *Bayesian network*, *M-separation*, and *Separation of powers*. Our automatic method labels *Bayesian network* as 1, since it is the top-ranked search result. It labels *M-separation* as 0.5, since it has the common category *Graphical models*, and it labels *Separation of powers* as 0.

[3] Three additional kinds of searches were also tried, including the basic query, searching with the "site:" restrictor, and searching on Wikipedia itself, but these had lower precision.

[4] This was the top-splitting rule learned by a decision tree classifier trained on the results of the four methods of search we tested. However, the decision tree was less good at predicting 0.5 and 0 annotations, leading us to manually produce the rules described.

While this is a simple method for predicting the relevance of Wikipedia results, it has a 0.7 correlation ($p < 0.001$) with the average of the human annotators – nearly as high as the inter-annotator agreement. We sampled an additional 30 Metacademy concepts for annotation by one of the authors as a validation set. The predictions had a 0.72 correlation with the human annotations for the validation set. Therefore, we used the results of this simple method for finding related Wikipedia articles.

We searched for 487 Metacademy concepts on DuckDuckGo and used our semi-automatic method for the rest of the unannotated matches (results of 487, minus 100 queries). Focusing on matches labeled 0.5 or 1, we retrieved 776 distinct Wikipedia articles (1,091 total matches). For 1,208 learning pairs we get 3,419 pairs of Wikipedia concepts.

Our semi-automatic mapping of Metacademy concepts to Wikipedia is being released at https://doi.org/10.6084/m9.figshare.7799774 and can be used to evaluate other work on prerequisite discovery. Our method for mapping concepts to Wikipedia's semantic space is general, and we expect it can be applied to gold-standard concept graphs that may exist for other domains.

3.4 Final Data Sets

The final data sets used for the prerequisite prediction experiments described in the next section consist of the entries from the aggregated Wikipedia clickstream data, where both articles belong to the set of concepts from the CMU or Metacademy prerequisites.

For the Metacademy data, we also compute sets of gold-standard prerequisites based on transitive closure. The transitive closure deals with the fact that, much as concepts can be enumerated at different levels of granularity, prerequisite relations may include intermediate dependent concepts or ignore them, e.g., we can say that *Hidden Markov model* depends on *Stochastic process* or we can say that *Hidden Markov Model* depends on *Markov chain*, which depends on *Stochastic process*. To compute the transitive closure, if (C_1, C_2) and (C_2, C_3) are prerequisite pairs, then we consider (C_1, C_3) as prerequisites as well.

Since we are dealing with a binary classification problem, the supervised learner needs some number of examples for both classes (prerequisite and not). In prerequisite relation identification, it is typically easy to find negative examples, while positive examples are scarce. Using the transitive closure can help for domains where we do not have enough labeled positive examples, ameliorating the problem of class imbalance.

While the CMU data set contains both positive and negative labeled pairs of concepts, Metacademy only contains positive examples, and we require some negative samples for our final data set. As a simple technique for generating negative samples, we make a closed-world assumption: Any example of navigation in the clickstream involving a concept from the domain that is not known to have a prerequisite relationship is taken as a negative example.

The original Metacademy data set is referred to as MA and the transitive closure, consisting of 14,633 pairs of concepts, as MA-TR. We use sub-sampling of majority class to create MA-bal and MA-bal-TR, which are balanced data sets – ones where there are the same number of observations for each class. The class distributions for the final Metacademy data sets are given in Table 1.

Table 1: Class distributions of Metacademy prerequisite data

Data Set	Original MA	Balanced MA-bal	Transitive closure Original MA-TR	Transitive closure Balanced MA-bal-TR
Prerequisites	10%	50%	18%	50%
Non-prereq.	90%	50%	82%	50%

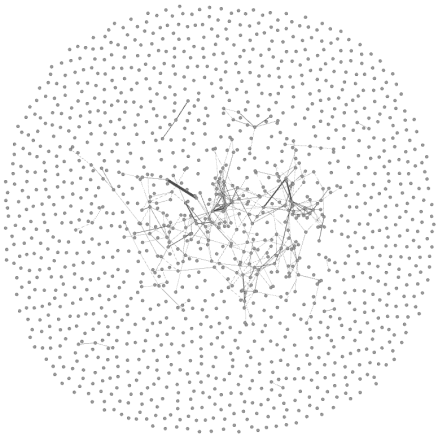

Figure 2: Visualizing positive prerequisite edges associated with Metacademy learning pairs (data set MA)

The graph of positive examples from MA is visualized in Figure 2, using the Yifan Hu proportional layout algorithm [8] in Gephi [1]. The large connected subgraph includes most of the relevant Wikipedia pages. Manual inspection of nodes in the graph shows nodes in the small subgraphs are mostly irrelevant results from the automatic matching using a search engine.

4 METHOD

To predict prerequisites, we train supervised classifiers on a set of features defined using the Wikipedia clickstream data for the concepts of interest. In this section, we define the features and discuss our choice of classifiers.

4.1 Features

We focused on user navigation–based features, excluding those related to the content of the articles. We create a directed, weighted graph from the clickstream matches, where the weight of an edge from article A to article B is the total number of clicks. We started with a few predictive features defined from this graph and incrementally added more until we achieved satisfying results. The final features are:

- *Weight.* Total number of clicks from article A to B.
- *Backward weight.* Total number of clicks from B to A.
- *Sum.* Sum of *weight* and *backward weight*.
- *Diff.* Absolute difference between *weight* and *backward weight* features.
- *Sum of all transitions.* Sum of the weights of all outgoing edges from A.

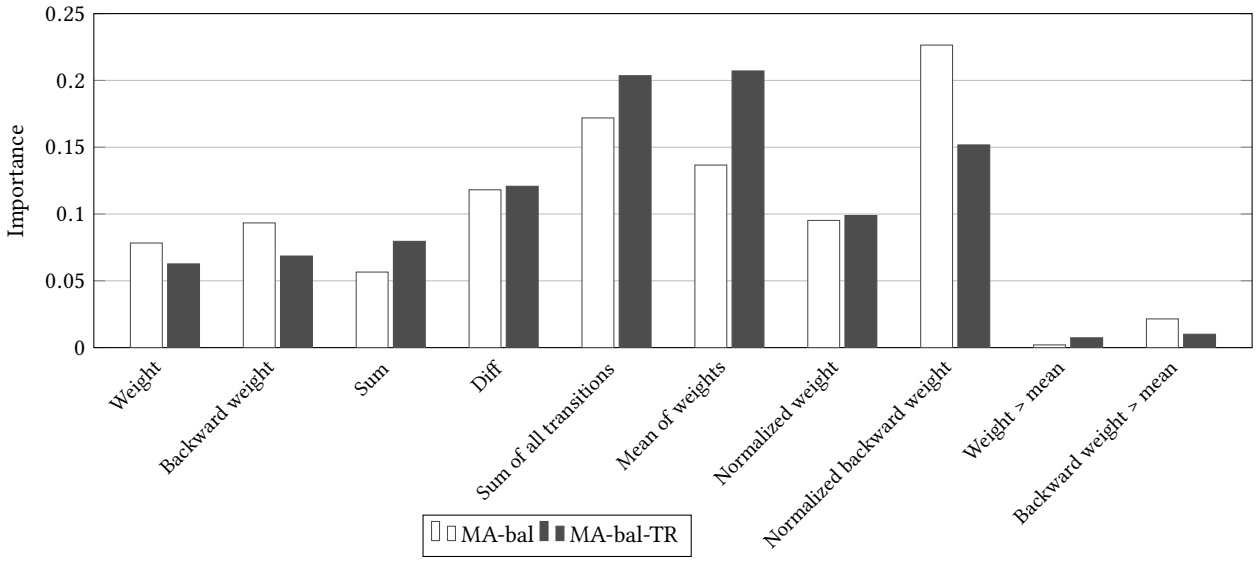

Figure 3: Feature importance for decision tree on MA-bal

- *Mean of weights.* Average of the weights of all outgoing edges from A.
- *Normalized weight.* Weight divided by sum of transitions.
- *Normalized backward weight.* The backward weight divided by the sum of all transitions for B.
- *Weight greater than mean.* A binary feature indicating whether the *weight* is greater than the *mean of weights*.
- *Backward weight greater than mean.* A binary feature indicating whether the *backward weight* is greater than B's *mean of weights*.

The intuition behind our use of backward navigation features (from an article to its prerequisites) is the idea that users who do not understand concept B will then study concept A if it will help them.

There is a low likelihood that we will have clickstream data (a non-trivial amount of direct navigation in the recorded months) for all pairs of articles in the gold-standard data sets. To improve the coverage of concepts, we also computed the maximum potential navigation (PN) on paths with one or two intermediate nodes. For one intermediate node, this is computed as:

$$\mathrm{PN}_1(A, B) = \max_C[\min(\mathrm{weight}[A \to C], \mathrm{weight}[C \to B])]$$

We did not consider any temporal properties of the available data, e.g., which articles existed or were connected by links for which months. Rather, we summed the clickstream counts for each pair across all available months.

4.2 Classifiers

For each data set, we trained binary classifiers using all of the features, using the scikit-learn [14] package in Python. In initial experiments, we used logistic regression and decision tree classifiers and then added AdaBoost and Gaussian naïve Bayes to compare the performance. We chose logistic regression as a classic classifier suitable for binary classification. We also used Gaussian naïve Bayes since we are dealing with continuous features and naïve Bayes is a simple algorithm used as an initial classification in many experiments. Decision tree is used since it can handle class imbalance better than other classical classifiers and can also produce human understandable rules. AdaBoost is a popular ensemble-learning method which usually gives more robust models. We set the maximum number of estimators in AdaBoost as 200 and use a decision tree with maximum depth equal to three as base estimators. We used these classifiers with five-fold cross validation and used 80% of the data set as training and 20% for testing.

5 RESULTS

Our goal is to label a pair of concepts as a prerequisite or not a prerequisite. We measure performance of classifiers on the Metacademy and CMU prerequisite data in terms of the accuracy, precision, recall, and F_1 scores. Accuracy shows the average performance of each classifier as the fraction of the pairs that have been successfully labeled.

$$\text{Accuracy} = \frac{\text{Correct predictions (both classes)}}{\text{All samples}}$$

Precision is the fraction of correctly labeled pairs.

$$\text{Precision} = \frac{\text{Prereq. pairs} \cap \text{Retrieved pairs}}{\text{Retrieved pairs}}$$

Recall calculates what fraction of all prerequisites are identified.

$$\text{Recall} = \frac{\text{Prereq. pairs} \cap \text{Retrieved pairs}}{\text{Prereq. pairs}}$$

In Tables 2 and 3, we present the performance of our classifiers for the original and transitive closure sets of prerequisite relations from Metacademy. The AdaBoost and decision tree classifiers, trained

Table 2: Performance of predictions on the Metacademy datasets MA and MA-bal

	MA				MA-bal			
	Acc.	Prec.	Rec.	F_1	Acc.	Prec.	Rec.	F_1
Random Classifier	0.50	0.10	0.50	0.17	0.50	0.50	0.50	0.50
Lazy Conservative Classifier	0.90	0	0	0	0.50	0	0	0
Logistic Regression	0.91	0.03	0.01	0.01	0.55	0.61	0.37	0.46
Gaussian Naïve Bayes	0.89	0.17	0.10	0.12	0.74	0.18	0.29	0.22
Decision Tree	**0.93**	0.61	**0.64**	**0.62**	**0.81**	0.78	0.77	0.77
AdaBoost	0.92	**0.69**	0.13	0.22	0.80	**0.80**	**0.78**	**0.80**

Table 3: Performance of predictions on the Metacademy datasets with transitive closure, MA-TR and MA-bal-TR

	MA-TR				MA-bal-TR			
	Acc.	Prec.	Rec.	F_1	Acc.	Prec.	Rec.	F_1
Random Classifier	0.50	0.18	0.50	0.26	0.50	0.50	0.50	0.50
Lazy Conservative Classifier	0.82	0	0	0	0.50	0	0	0
Logistic Regression	0.88	0.12	0.01	0.02	0.56	0.60	0.57	0.59
Gaussian Naïve Bayes	0.86	0.19	0.06	0.09	0.54	0.67	0.16	0.26
Decision Tree	**0.91**	0.62	**0.65**	**0.63**	**0.78**	0.77	**0.79**	**0.78**
AdaBoost	0.90	**0.73**	0.17	0.28	**0.78**	**0.78**	0.78	**0.78**

over balanced data sets (MA-bal and MA-bal-TR) produce the best results achieving precision, recall, and F_1 scores of around 80%.

Since negative examples constitute 90% of data set MA, a lazy conservative classifier can predict with 90% accuracy simply by predicting every pair of concepts not to be prerequisites. (This is known as the accuracy paradox.) We include this lazy classifier and a random classifier as baselines. In the Metacademy prerequisite data sets MA and MA-TR, the performance of logistic regression and Gaussian naïve Bayes classifiers could not beat the baseline lazy classifier. As expected, logistic regression is highly sensitive to class imbalance, and it ignores the minority class. This is due to the cost function and update rule of logistic regression, where a good model for the majority class can minimize the cost function. The naïve Bayes classifier gives poor performance because we are using a basic version of naïve Bayes in which the class prior biases the predictions toward the majority class.

Performance can be improved by using a classifier that better handles class imbalance, such as decision tree. Decision tree is resilient against class imbalance because it selects the splitting rules based on information gain (or Gini index), which can force both classes to be addressed. The scikit-learn library uses an optimized version of the CART algorithm [2], which tries to induce a tree with largest information gain at each node. Sub-sampling the majority class can also help our classifier to boost the precision and recall in the balanced data sets (MA-bal and MA-bal-TR).

In our experiments, using the transitive closure did not significantly change the performance. This is the case because our gold standard had enough positive pairs and even finding matched clickstream records for a subset of them sufficed to train a good classifier. Especially in the original data set (MA), using transitive closure (MA-TR) could not beat the class imbalance problem in our data set.

Looking at the learned decision tree for MA, the top splitting feature is *Normalized backward weight*. This supports our hypothesis that learners will read an article and, if they do not understand the concept, will navigate to its prerequisite concepts. Figure 3 shows the importance of different features for a decision tree classifier based on Gini index computed on MA-bal and MA-bal-TR.

Table 4 shows the coverage of the Wikipedia clickstream data on different domains of the CMU data set and Metacademy. Including intermediate nodes helps to increase the coverage especially for the CMU data set. The average coverage using direct links is 51%, which increases to 70% while using one intermediate node and 76% while using two intermediate nodes. Note that there are multiple reasons there may not be a match in the clickstream. For the Metacademy data, 15% of the concepts have no matching Wikipedia articles. Another possibility is that there is no direct hyperlink between the Wikipedia articles; using the Wikipedia API, we find that for only 28% of these pairs we do not find in the clickstream there is an associated link in Wikipedia.

Table 5 shows the performance of a decision tree classifier for the CMU data set over the covered concepts. The performance of the classifiers followed the pattern observed for the Metacademy data sets, so we only report the best results from the decision tree, although AdaBoost performed similarly. It is difficult to directly compare our results to previous work using the CMU prerequisite data set. Talukdar and Cohen [15] do not explain the details of their in-domain training approach (since their main focus was out-of-domain), so we cannot mimic the details of their experiment for a fair comparison. However, for the subset of the data set covered by the Wikipedia clickstream, our predications are more accurate than their maximum entropy classifier, as shown in Figure 4. Liang et al. [10] also compare their performance on the CMU data set,

Table 4: Coverage for the CMU and Metacademy data sets

Coverage with	Newton's Laws	Public-key Cryptography	Global Warming	Parallel Postulate	Meiosis	Metacademy
Direct link	41%	63%	55%	36%	58%	18%
1 intermediate node	55%	72%	78%	61%	82%	30%
2 intermediate nodes	60%	79%	88%	68%	86%	31%

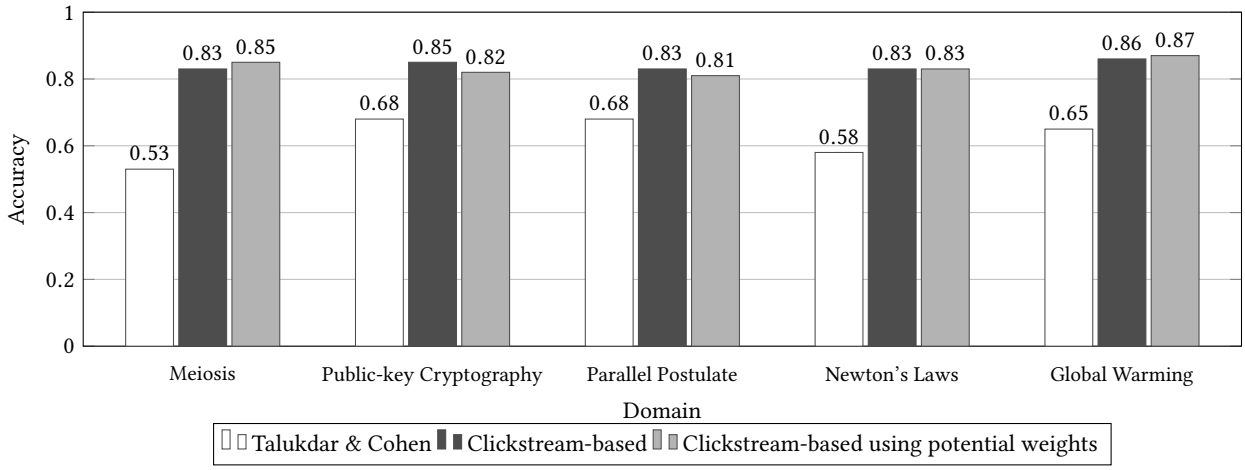

Figure 4: Comparison of our accuracy with Talukdar and Cohen [15] for the subset of the CMU data set covered by the Wikipedia clickstream

Table 5: Performance of our decision tree classifier on the five domains of the CMU data set

	Acc.	Prec.	Recall	F_1 Score
Newton's Laws	0.83	0.81	0.87	0.84
Public-key Cryptography	0.82	0.81	0.85	0.83
Meiosis	0.85	0.84	0.86	0.85
Parallel Postulate	0.81	0.82	0.82	0.82
Global Warming	0.87	0.86	0.88	0.87

using out of domain training. Since our approach uses in-domain training, we cannot directly compare our results to theirs.

6 CONCLUSION

To help learners follow a coherent path through a knowledge domain, we want to infer when one concept is a prerequisite of another. We have described a new approach to this problem that exploits the "clickstream" of human navigation among articles on Wikipedia. In particular, we find that an important feature is the backward navigation from a more advanced concept to one of its prerequisites. We evaluate our classification methods over an existing gold standard of prerequisites crowdsourced by researchers at CMU and over a new data set of expert-generated prerequisites from Metacademy that we map into Wikipedia's semantic space. Training decision tree and AdaBoost classifiers over balanced datasets, we obtain precision, recall and F_1 scores of around 80% on this new Metacademy dataset and F_1 measures from 82% to 87% on the CMU dataset. We hope that these new techniques based on navigation data will enable self-directed learners to take a greater advantage of the vast amounts of learning materials available on the Web.

ACKNOWLEDGMENTS

This work is supported by the National Institutes of Health under Grant U24ES026465. Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the National Institutes of Health.

REFERENCES

[1] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. 2009. Gephi: An Open Source Software for Exploring and Manipulating Networks. In *Proceedings of the International AAAI Conference on Weblogs and Social Media* (*ICWSM*). AAAI Press, Palo Alto, CA, USA, 361–2. https://www.aaai.org/ocs/index.php/ICWSM/09/paper/view/154

[2] Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. 1984. *Classification and Regression Trees.* Chapman & Hall / CRC, New York.

[3] Evgeniy Gabrilovich and Shaul Markovitch. 2007. Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence* (*IJCAI*). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1606–11.

[4] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning.* MIT Press, Cambridge, MA, USA. https://www.deeplearningbook.org.

[5] Jonathan Gordon, Stephen Aguilar, Emily Sheng, and Gully A. P. C. Burns. 2017. Structured Generation of Technical Readings Lists. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications* (*BEA*). Association for Computational Linguistics, Copenhagen, Denmark, 261–70. https://doi.org/10.18653/v1/W17-5029

[6] Jonathan Gordon, Linhong Zhu, Aram Galstyan, Prem Natarajan, and Gully A. P. C. Burns. 2016. Modeling Concept Dependencies in a Scientific Corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics* (*ACL*). Association for Computational Linguistics, Berlin, Germany, 866–75. https://doi.org/10.18653/v1/P16-1082

[7] Roger Grosse and Colorado Reed. 2013. Metacademy. https://metacademy.org.

[8] Yifan F. Hu. 2005. Efficient and high quality force-directed graph drawing. *The Mathematica Journal* 10 (2005), 37–71. Issue 1.

[9] Daniel Lamprecht, Denis Helic, and Markus Strohmaier. 2015. Quo Vadis? On the Effects of Wikipedia's Policies on Navigation. In *Wikipedia, a Social Pedia: Research Challenges and Opportunities: Workshop at the Ninth International Conference on Web and Social Media*. AAAI Press, Palo Alto, CA, USA, 64–6. https://aaai.org/ocs/index.php/ICWSM/ICWSM15/paper/view/10653

[10] Chen Liang, Zhaohui Wu, Wenyi Huang, and C. Lee Giles. 2015. Measuring Prerequisite Relations Among Concepts. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, 1668–74. https://doi.org/10.18653/v1/D15-1193

[11] Chen Liang, Jianbo Ye, Shuting Wang, Bart Pursel, and C. Lee Giles. 2018. Investigating Active Learning for Concept Prerequisite Learning. In *Proceedings of the Eighth Symposium on Educational Advances in Artificial Intelligence* (*EAAI*). AAAI Press, Palo Alto, CA, USA, 7913–9.

[12] Yilun Lin, Bowen Yu, Andrew Hall, and Brent Hecht. 2017. Problematizing and Addressing the Article-as-Concept Assumption in Wikipedia. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work and Social Computing* (*CSCW*). ACM, New York, NY, USA, 2052–67. https://doi.org/10.1145/2998181.2998274

[13] Carlo De Medio, Fabio Gasparetti, Carla Limongelli, Filippo Sciarrone, and Marco Temperini. 2016. Automatic Extraction of Prerequisites Among Learning Objects Using Wikipedia-Based Content Analysis. In *Proccedings of the 13th International Conference on Intelligent Tutoring Systems* (*ITS*), Alessandro Micarelli, John Stamper, and Kitty Panourgia (Eds.). Springer International Publishing, Switzerland, 375–81. https://doi.org/10.1007/978-3-319-39583-8_44

[14] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–30.

[15] Partha Pratim Talukdar and William W. Cohen. 2012. Crowdsourced Comprehension: Predicting Prerequisite Structure in Wikipedia. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*. Association for Computational Linguistics, Montréal, Canada, 307–15.

[16] Shuting Wang, Alexander Ororbia, Zhaohui Wu, Kyle Williams, Chen Liang, Bart Pursel, and C. Lee Giles. 2016. Using Prerequisites to Extract Concept Maps from Textbooks. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management* (*CIKM*). ACM, New York, NY, USA, 317–26. https://doi.org/10.1145/2983323.2983725

[17] Ellery Wulczyn and Dario Taraborelli. 2017. Wikipedia Clickstream. https://doi.org/10.6084/m9.figshare.1305770.v22