# Crowdsourced Comprehension: Predicting Prerequisite Structure in Wikipedia

**Partha Pratim Talukdar**
Machine Learning Department
Carnegie Mellon University
ppt@cs.cmu.edu

**William W. Cohen**
Machine Learning Department
Carnegie Mellon University
wcohen@cs.cmu.edu

## Abstract

The growth of open-access technical publications and other open-domain textual information sources means that there is an increasing amount of online technical material that is in principle available to all, but in practice, incomprehensible to most. We propose to address the task of helping readers comprehend complex technical material, by using statistical methods to model the "prerequisite structure" of a corpus — i.e., the semantic impact of documents on an individual reader's state of knowledge. Experimental results using Wikipedia as the corpus suggest that this task can be approached by crowdsourcing the production of ground-truth labels regarding prerequisite structure, and then generalizing these labels using a learned classifier which combines signals of various sorts. The features that we consider relate pairs of pages by analyzing not only textual features of the pages, but also how the containing corpora is connected and created.

## 1 Introduction and Motivation

Nicholas Carr has argued in his recent popular book "The Shallows" that existing Internet technologies encourage "shallow" processing of recent and popular information, at the expense of "deeper", contemplative study of less immediately-accessible information (Carr, 2011) . While Carr's hypothesis is difficult to formalize rigorously, it seems intuitively plausible. For instance, user-generated content from Twitter and Facebook is mainly comprised of short, shallow snippets of information. Most current research in AI (and more broadly in computer science) does not seem likely to reverse this trend: e.g., work in crowdsourcing has concentrated on tasks that can be easily decomposed into small pieces, and much current NLP research aims at facilitating short-term "shallow" goals, such as answering well-formulated questions (e.g., (Kwok et al., 2001)) and extracting concrete facts (e.g., (Etzioni et al., 2006; Yates et al., 2007; Carlson et al., 2010)). This raises the question: what can AI do to facilitate deep, contemplative study?

In this paper we address one aspect of this larger goal. Specifically, we consider automation of a novel task—using AI methods to facilitate the "deep comprehension" of complex technical material. We conjecture that the primary reason that technical documents are difficult to understand is lack of modularity: unlike a self-contained document written for a general reader, technical documents require certain background knowledge to comprehend—while that background knowledge may also be available in other on-line documents, determining the proper sequence of documents that a particular reader should study is difficult.

We thus formulate the problem of comprehending technical material as a probabilistic planning problem, where reading a document is an operator that will probabilistically change the state of knowledge $K(u,t)$ of a user $u$ at time $t$, in a manner that depends on $u$'s prior knowledge $K(u,t-1)$. Solving this task requires, among other things, understanding the effect of reading individual documents $d$ — specifically, the concepts that are explained by $d$, and the concepts that are *prerequisites* for comprehending $d$. This paper addresses this problem. In particular, we consider predicting whether one page in Wikipedia is a prerequisite of another.

More generally, we define the "prerequisite struc-



*The 7th Workshop on the Innovative Use of NLP for Building Educational Applications*, pages 307–315,
Montréal, Canada, June 3-8, 2012. 

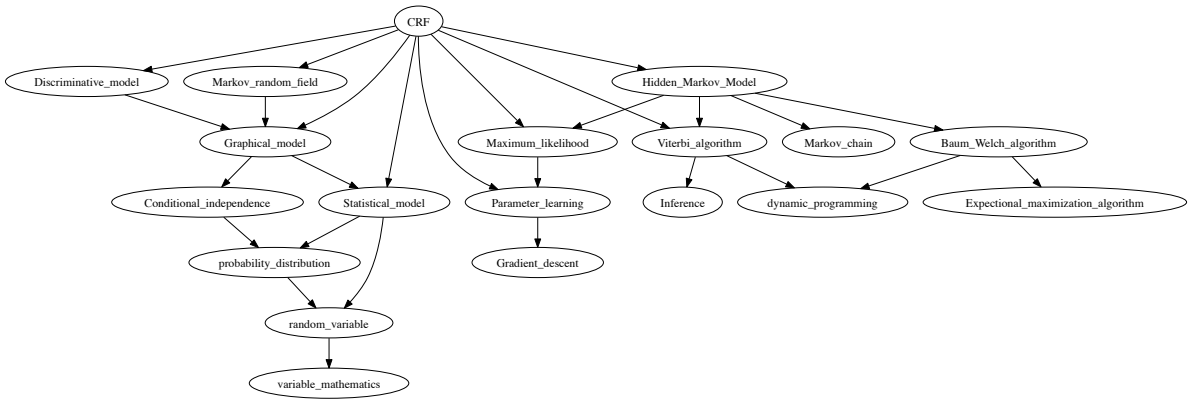


Figure 1: The prerequisite structure rooted at the page "Conditional Random Fields", omitting nodes that would already be known a typical CS graduate student.

| | | |
|---|---|---|
| Variable (Mathematics) | Random Variable | Probability Distribution |
| Conditional Independence | Statistical Model | Graphical Model |
| Discriminative Model | Markov Random Field | |
| Gradient Descent | Parameter Learning | Maximum Likelihood |
| Inference | Dynamic Programming | Viterbi Algorithm |
| Markov Chain | Expectation Maximization Algorithm | Baum Welch Algorithm |
| Hidden Markov Model | CRF | |

Figure 2: A plan for comprehending "Conditional Random Fields" (to be read left-to-right, top-to-bottom). Horizontal lines indicate breaks between independent sections of the subgraph.

ture" for a corpus as a graph, where nodes are concepts to comprehend, and a directed edge $d \rightarrow d'$ corresponds to the assertion "understanding $d'$ is a prerequisite to understanding $d$". For Wikipedia, we assume a one-to-one correspondence between document titles and concepts explicated by (i.e., postconditions of) these documents. Figure 2 presents a small example of a prerequisite structure, and indicates how it might be used to construct a plan for comprehending a specific concept.

Focusing on Wikipedia has several advantages. First, it is densely linked, and hence a document $d$ will likely be linked directly to any prerequisite page $d'$. (However, not all hyperlinks will indicate a prerequisite.) Second, Wikipedia's standardized format makes textual analysis easier. Finally, there is a great deal of social information available about how documents are used by the Wikipedia community. These properties make it easy for us to explore the informativeness of different types of information with respect to predicting prerequisite structure.

Our overall plan for producing a prerequisite structure for a corpus is first, to use crowdsourcing approaches to obtain a subset of the prerequisite structure; and second, to extrapolate this structure to the entire corpus using machine learning. Below, we first describe datasets that we have collected, based on five technical concepts in Wikipedia from five different fields. We then outline the specifics of our procedure for annotating prerequisite structure, using Amazon's Mechanical Turk, and demonstrate that meaningful signals about prerequisite structure can be obtained using a classifier that exploits several sources: graph analysis of Wikipedia's link graph; graph analysis of a bipartite graph relating Wikipedia pages to Wikipedians that have edited these pages; and textual analysis. We complete our experimental analysis of the prerequisite-structure prediction task by discussing and evaluating the degree to which prerequisite-structure prediction is domain-independent, and the degree to which different subareas of Wikipedia (e.g., biology *vs* computer science) require different predictors.

After discussing related work, we return in the concluding remarks to the overarching goal of facilitating comprehension, and discuss the relation-

| Target Concept | #Nodes | #Edges | #Edits |
|---|---|---|---|
| Global Warming | 19,170 | 501,608 | 1,490,967 |
| Meiosis | 19,811 | 444,100 | 880,684 |
| Newton's Laws of Motion | 15,714 | 436,035 | 795,988 |
| Parallel Postulate | 14,966 | 363,462 | 858,785 |
| Public-key cryptography | 16,695 | 371,104 | 1,003,181 |

Table 1: Target concepts used in the experiments.

ship of the current study to these goals. Specifically we note that facilitating comprehension also requires understanding a user's goals, and her initial state of knowledge, in addition to understanding the prerequisite structure of the corpus. We also discuss the relationship between planning and prerequisite-structure prediction and suggest that use of appropriately robust planning methods may lead to good comprehension plans, even with imperfectly predicted prerequisite structure.

## 2 Experiments

As discussed above, we focus in this paper on predicting prerequisite structure in Wikipedia. While most Wikipedia pages are accessible to a general reader, there are many pages that describe technical concepts, such as "conditional random fields", "cloud radiative forcing", and "Corticotropin-releasing factor". Most of these technical pages are not self-contained: for instance, to read and comprehend the page on "conditional random fields", one will have to first understand "graphical model", and so on, as suggested by Figure 1. In this section, we evaluate the following questions:

- Can we train a statistical classifier for prerequisite classification in a target domain, where the classifier is trained on out of domain (i.e., non-target domain) data annotated using Amazon Mechanical Turk service?
- What are the effects of different types of signals on the performance of such a classifier?
- How does out of domain training compare to in domain training?

### 2.1 Experimental Setup

For our experiments, we choose five targets from differing areas for experimentation, listed in Table 1. Several of the techniques we used are based on graph analysis. The full graphs associated with Wikipedia are unwieldy to use for experimentation because of their size: therefore, for each target concept, we extracted a moderate-sized low-conductance subgraph of Wikipedia's link graph containing the target, using a variant of the PageRank-Nibble algorithm (Andersen et al., 2006).[1]. As parameters we used $\alpha = 0.15$ and $\epsilon = 10^{-7}$, yielding graphs with approximately 15-20,000 nodes and 350-500,000 edges each. We also collected the edit history for each page in every subgraph forming a second graph for each sub-domain [2]. On average, each page from these subgraphs had been edited about 20 times, by about 8 unique editors. Details are given in Table 1.

For classification, we used a Maximum Entropy (MaxEnt) classifier. Given a pair of Wikipedia pages $x = (d, d^{'})$ connected by a directed edge (hyperlink) from $d$ to $d^{'}$, the classifier will predict with probability $p(+1|x)$ whether the main concept in page $d^{'}$ is a prerequisite for the main concept in page $d$. The classifier has the form

$$p(y|x) = \frac{\exp(\mathbf{w} \cdot \phi(x, y))}{\sum_{y' \in Y} \exp(\mathbf{w} \cdot \phi(x, y'))},\ y \in Y = \{-1, +1\}$$

where $\phi(x, y)$ is a feature function which represents the pair of pages $x = (d, d^{'})$ in a high dimensional space, and $\mathbf{w}$ is the parameter vector of the classifier which is estimated from training data. We use the Mallet package[3] to train and evaluate classifiers. For the experiments in this paper, we shall exploit the following types of features:

**WikiHyperlinks**: Features include the random walk with restart (RWR) score (Tong et al., 2006) of the target concept page $d^{'}$ starting from the source page $d$. Additional features include the PageRank score of the target and source pages.

[1] Specifically, we used the "ApproximatePageRank" method from (Andersen et al., 2006) to find a set of nodes $S$ containing a low-conductance subgraph, but did not prune $S$ to find the lowest-conductance subgraph of it with a "sweep". The version of Wikipedia's link graph we used was DBPedia's version 3.7 (Auer et al., 2007)

[2] Specifically, a bipartite graph connecting pages and editors. We used a version of Wikipedia's edit history extracted by other researchers (Leskovec et al., 2010), discarding edits marked as "minor" by the editor.

[3] Mallet package: http://mallet.cs.umass.edu/

| Domain | Time (s) / Evaluation | Worker / HIT | # HITs | $\kappa$ |
|---|---|---|---|---|
| Meiosis | 38 | 3 | 400 | 0.50 |
| Public-key Cryp. | 26 | 3 | 200 | 0.63 |
| Parallel Postulate | 41 | 3 | 200 | 0.55 |
| Newton's Laws | 20 | 5 | 400 | 0.47 |
| Global Warming | 14 | 5 | 400 | 0.56 |
| Average | 27.8 | - | - | 0.54 |

Table 2: Statistics about the Gold-standard data prepared using Amazon Mechanical Turk. Also shown are the averaged $\kappa$ statistics-based inter-annotator agreement in each domain. The last row corresponds to the $\kappa$ value averaged across all five domains.

**WikiEdits**: This includes one feature—the analogous RWR score on the graph of edit information.

**WikiPageContent**: Features in this category are derived from the contents of the two Wikipedia pages $d$ and $d^{'}$. Examples include: the category identity of the source page; the category identity of the target page; whether the titles of $d^{'}$ and $d$ are mentioned in the first sentence of $d$; the name of the first section in $d$ which contains a link to $d^{'}$; whether there is any overlap in categories between the two pages; whether $d$ is also linked from $d^{'}$; and the log of the number of times $d'$ is linked form $d$. We use the JWPL library (http://jwpl.googlecode.com) for efficient and structured access to Wikipedia pages from a recent dump obtained on Jan 4, 2012.

### 2.1.1 Gold-standard Annotation from Mechanical Turk[4]

In order to evaluate different prerequisite classification systems and also to train the MaxEnt classifier, we collected gold prerequisite decisions using Amazon Mechanical Turk (AMT). Since preparing annotated gold data for entire graphs in Table 1 would be prohibitively expensive, we used the following strategy to sample a smaller subgraph from the larger domain-specific subgraph, which in turn will be used for training and evaluation purposes. Preliminary investigation suggested that most of the pages in the prerequisite structure rooted at a target concept $d$ are connected to $d$ via many short hyperlink paths. Hence, for each target domain, we first selected the top 20 nodes with highest RWR scores, relative to the target concept, in the subgraph for that target concept (as listed in Table 1.) We then sampled a total of 400 edges from these selected nodes, with outgoing edges from a node sampled with a frequency proportional to its RWR score. Thus, using this strategy, we selected up to 400 pairs of pages $(d, d^{'})$, where each pair has a hyperlink from $d$ to $d'$.

Classification of a pair of hyperlinked Wikipedia pages $(d, d^{'})$ into one of the four following classes constituted a Human Intelligence Task (HIT): (1) $d^{'}$ is a prerequisite of $d$; (2) $d$ is a prerequisite of $d^{'}$; (3) the two pages are unrelated; (4) Don't know. Subsequently, based on the feedback from the workers, a fifth option was also added: the two concepts are related, but they don't have any prerequisite relationship between them. Based on the available workers and turnaround time, the number of assignments per HIT (i.e., number of unique workers assigned to a particular HIT) was either 3 or 5; and the number of HITs used was either 200 or 400. Depending on the hardness of domain and availability of workers opting to work on a domain, reward per HIT assignment was varied from \$0.02 (for Global Warming and Newton's Laws) to \$0.08 (for Public-key Cryptography, Meiosis and Parallel Postulate). This data collection stage spanning all five domains was completed in about a week at a total cost of \$278. Statistics about the data are presented in Table 2[5].

Starting with the AMT data collected as above, we next created a binary-labeled training dataset, where each instance corresponds to a pair of pages. We ignored all "Don't Know" labels, treated option (1) above as vote for the corresponding prerequisite edge, and treated all other options as votes against. We then assigned the final label for a node pair using majority vote (breaking ties arbitrarily).

### 2.1.2 Consistency of labels

In contrast to standard setup of gold data preparation where a single annotator is guaranteed to provide feedback on every instance, the situation in case of Mechanical Turk-based annotation is different, as the workers are at liberty to choose the HITs (or instances) they want to work on. This makes

[4]Amazon Mechanical Turk: http://mturk.amazon.com

[5]The dataset is available upon request from the authors.

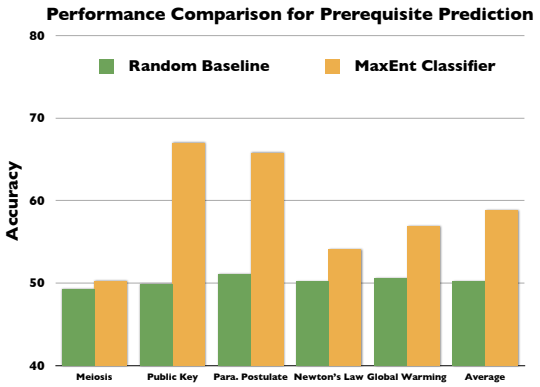


Figure 3: Comparison of performance between the MaxEnt classifier (right bar in each group) against a random baseline (left bar in each group) in all five domains. On average, the MaxEnt classifier results in an 8.6% absolute improvement in accuracy.

standard $\kappa$ statistics-based inter-annotator computation (Fleiss, 1981) inapplicable in the current setting. We circumvented this problem by first selecting all workers with at least 100 feedbacks, and then computing pairwise $\kappa$ statistics between all pairs of these frequent workers. These $\kappa$ statistics were averaged across each domain, and also averaged across all domains. The results, also shown in Table 2, show moderate agreement (recall that $\kappa = 0$ indicates no correlation). We are encouraged to observe that moderate level of agreement is possible even in this setting, where there is no control over worker background and quality. We next explore whether this level of agreement is sufficient to train statistical classifiers.

## 2.2 Prerequisite Classification

In this section, we explore whether it is possible to train a MaxEnt classifier to determine prerequisite structure in a target domain, with the training performed in "leave one domain out" manner, where the training data originates from domains other than the target domain. For example, for classifications in the target domain, say "Global Warming", we train the classifier with annotated data from the remaining four domains (or whatever domains are available). We note that training on "out of domain", if it is successful, has several benefits. First, a successful training strategy in this setup removes any need to have labeled data in each target domain of interest, which is particularly relevant as labeled data is expensive to prepare. Second, a classifier trained just once can be repeatedly used across multiple domains without requiring retraining.

Accuracies of MaxEnt classifiers trained using the "leave one domain out" strategy are shown in Figure 3; we report the test accuracy on each target domain, as well as the average across domains. Performance of a random classifier is presented as a baseline. Classes in the train and test sets were balanced by oversampling the minority class. From Figure 3, we observe that it is indeed possible to train prerequisite classifiers in an out of domain setting, using data from the Amazon Mechanical Turk service; on average, the classifier outperforms the random baseline with 8.6% absolute improvement in classification accuracy. We also experimented with other rule-based classifiers[6], and in all cases, the trained MaxEnt classifier outperformed these baselines. Although more sophisticated training strategies and more clever feature engineering would likely yield further improvements, we find it encouraging that even a relatively straightforward classification technology along with a basic set of features was able to achieve significant improvement in performance on the novel task of prerequisite prediction.

## 2.3 Feature Ablation Experiments

The MaxEnt classifier evaluated in the previous section had access to all three types of features: WikiEdits, WikiHyperLinks, and WikiPageContent, as described in the beginning of this section. In order to evaluate the contribution of each such signal, we created ablated versions of the full MaxEnt classifier which uses only one of these three subsets. We call these thee variants: MaxEnt-WikiEdits, MaxEnt-WikiHyperLinks, and MaxEnt-WikiPageContent, respectively. Average accuracies across all five domains comparing these three variants, in comparison to the Random baseline and the full classifier (MaxEnt-Full, as in previous section) are presented in Table 3. From this, we observe that all three variants perform better than the random baseline, with maximum gains achieved by the MaxEnt-WikiPageContent classifier, which uses page content-based features exclusively. We

[6]For example, classify $d'$ as a prerequisite for $d$ if $d'$ is linked from the first paragraph in $d$.

| System | Accuracy |
|---|---|
| Random | 50.22 |
| MaxEnt-WikiEdits | 51.62 |
| MaxEnt-WikiHyperlinks | 52.70 |
| MaxEnt-WikiPageContent | 57.84 |
| MaxEnt-Full | 58.82 |

Table 3: Comparison of accuracies (averaged across all five domains) of the full MaxEnt classifier with its ablated versions which use a subset of the features, and also the random baseline. The full classifier, which exploits all three types of signals (viz., WikiEdits, WikiHyperlinks, and WikiPageContent) achieves the highest performance.

| Domain | Wiki-Edits | Wiki-HyperLinks | WikiPage-Content | All |
|---|---|---|---|---|
| Meiosis | **5.4** | **2.4** | **0.3** | **1** |
| Public-key Crypto. | -0.7 | -1.8 | **15.1** | **17.1** |
| Parallel Postulate | **3.1** | **6.1** | **11.7** | **14.7** |
| Newton's Laws | -0.2 | **6.2** | **3.9** | **3.9** |
| Global Warming | -7.7 | **0.1** | **5.8** | **6.8** |

Table 4: Accuracy gains (absolute) relative to the Random baseline achieved by the full MaxEnt classifier as well as its ablated versions trained with three different subsets of the full classifier. Positive gains are marked in bold.

also note that the full classifier MaxEnt-Full, is able to effectively combine three types of signals improving performance even further. In Table 4, we present a per-domain breakdown of the gains achieved by these four classifiers over the random baseline. From this, we observe that the MaxEnt-WikiEdits classifier outperforms the random baseline only in 2 out of 5 domains. This might be due to the fact that the MaxEnt-WikiEdits uses uses only one feature—the RWR score of the target page relative to the source page on the Wikipedia edits graph. We hope that use of more discriminating features should further help this classifier. From Table 4, we also observe that MaxEnt-WikiHyperLinks is able to outperform the random baseline in 4 out of 5 cases, and the MaxEnt-WikiPageContent (as well as the full classifier) outperforms the random baseline in all 5 domains, sometimes with large gains (as in the case of Public-key Cryptography domain).

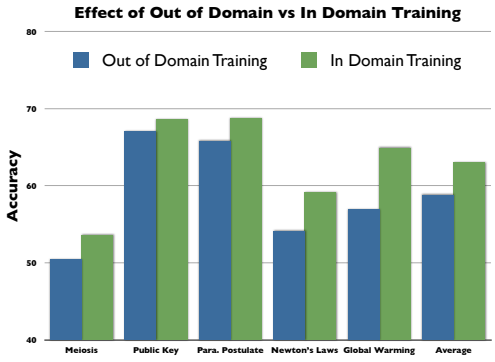


Figure 4: Accuracy comparison of out of domain (left bar in each group) and in domain training (right bar in each group) for the five domains. From this we observe that good generalization performance is possible even when there is no in domain training data available.

## 2.4 Effect of Out of Domain Training

All the classifiers evaluated in previous sections were trained in an out of domain setting, i.e., the training data originated from domains outside the domain in which the classifier is applied and evaluated. This has several benefits, as noted above. An alternative and more standard way to train classifiers is to have the training and evaluation data be from the same domain (below, the in-domain setting). While such a classifier will require labeled training from each domain of interest, it is nonetheless of interest to compare in-domain and out-of-domain learning. If there are substantive differences, this could be used to improve prerequisite-structure predictor in a subdomain (e.g., biology), or may suggest alternative training methods (e.g., involving transfer learning).

Motivated by this, for each domain, we compare the performances of the out-of-domain and in-domain classification performances. The results are shown in Figure 4. On average, we observe that the out-of-domain classifier is able to achieve 93% of the performance of the in-domain classifier. We note that this is encouraging for domain-independent prerequisite-structure prediction, as this suggests that for the prerequisite classification task, close to optimal (i.e., in-domain performance) is possible when the classifiers are trained in an out-of-domain setting.

## 3 Related Work

We believe the task of prerequisite structure prediction to be novel; however, it is clearly related to a number of other well-studied research problems.

In light of our emphasis on Wikipedia, a connection can be drawn between identifying prerequisites and measuring the semantic relatedness of concepts using Wikipedia's link structure (Yeh et al., 2009). We consider here a related but narrower question, namely whether an inter-page link will improve comprehension for a specific reader.

In the area of intelligent tutoring and educational data mining, recent research has looked at enriching textbooks with authoritative web content (Agrawal et al., 2010). Also, the problem of detecting prerequisite structure from differential student performance on tests has been considered (e.g., (Pavlik et al., 2008; Vuong et al., 2011)). Our proposal considers discovering prerequisite structure from text, rather than from exercises, and relies on different signals.

Research in adaptive hypermedia (surveyed elsewhere (Chen and Magoulas, 2005)) has goals similar to ours. Most adaptive hypermedia systems operate in narrow domains, which precludes use of some of the crowd-based signals we consider here. In this literature, a distinction is often made between "adaptability" (the ability for a user to modify a presentation of hypermedia) and "adaptivity" (the ability of a system to adapt to a user's needs.) In this framework, our project focuses on adding "adaptivity" to existing corpora via a prerequisite structure, and our principle contribution to this area is identifying techniques that learn to combine textual features and social, crowd-based signals in order to usefully guide comprehension.

Another related area is data-mining logs of Web usage, as surveyed by Pierrakos *et al* (Pierrakos et al., 2003). Our focus here is on facilitating a particular type of Web usage, comprehension, rather than more commonly-performed tasks like site navigation and purchasing.

A number of "open education" resources exist, in which information can be organized into sharable modules with known prerequisites between them (e.g., Connexions (Baraniuk, 2008)). We focus here on discovering prerequisite structure with machine-learning methods rather than simply encoding it. Similarly, a Wikimedia project[7] has developed infrastructure allowing a user to manually assemble Wikipedia pages into e-books. Our focus is on guiding the process of finding and ordering the sections of these books, not the infrastructure for generating them. We also note that one widely-used way for complex technical concepts to be broadly communicated is by writers or teams of writers, and previous researchers have investigated understanding how collaborative writers work (Noël and Robert, 2004), and even developed tools for collaborative writing (Zheng et al., 2006). Our work focuses on tools to empower readers, rather than writers.

## 4 Conclusion

In this paper, we motivated the goal of "crowdsourcing" the task of helping readers comprehend complex technical material, by using machine learning to predict prerequisite structure from not only document text, but also crowd-generated data such as hyperlinks and edit logs. While it is not immediately obvious that this task is feasible, our experiments suggest that relatively reliable features to predict prerequisite structure exist, and can be successfully combined using standard machine learning methods.

To achieve the broader goal of facilitating comprehension, predicting prerequisite structure is not enough. Another important subproblem is using predicted prerequisites to build a feasible plan. As part of ongoing work, we are exploring use of modern optimization methods (such as Integer Linear Programming) to compute "reading plans" that minimize a weighted linear combination of expected user effort and probability of plan "failure"[8].

We also plan to explore another major subproblem associated with facilitating comprehension—personalizing a reading plan. Clearly, even if $d'$ is a prerequisite for $d$, a user interested in $d$ need not first read a page explaining $d'$, if she already understands $d'$; instead, a reading plan based on prerequisite structure should be adjusted based on what is believed about the user's prior knowledge state. In

[7] See http://en.labs.wikimedia.org/wiki/Wiki_to_print, the "Wiki to Print" project.

[8] A plan "failure" means that the plan not actually satisfy all necessary prerequisites, leading to imperfect comprehension on the part of the reader after she executes the plan.

the context of Wikipedia comprehension, one possible signal for predicting an individuals' prior knowledge is the Wikipedia edit log: if we assume that editors tend to edit things they know, the edit log indicates which concepts tend to be jointly known, and hence collaborative-filtering methods might be able to more completely predict a user's knowledge given partial information about her knowledge—just as collaborative-filtering is often used now to extrapolate user preference's from knowledge of others' joint preferences.

Besides contributing to the goal of facilitating comprehension, we believe that the specific problem of predicting prerequisite structure in Wikipedia is a task of substantial independent interest. Prerequisite structure can be thought of as a sort of explanatory discourse structure, which is overlaid on a hyperlink graph; hence, scaling up our methods and applying them to all of Wikipedia would identify a canonical broad-coverage instance of such explanatory discourse. This could be re-used for other tasks much as lexical resources like WordNet are: for instance, consider identifying explanatory discourse in an external technical text (e.g., a textbook) by soft-matching it to the Wikipedia structure, using existing techniques to match the external text to Wikipedia (Agrawal et al., 2010; Mihalcea and Csomai, 2007; Milne and Witten, 2008).

## Acknowledgments

This research has been supported in part by DARPA (under contract number FA8750-09-C-0179), and Google. Any opinions, findings, conclusions and recommendations expressed in this paper are the authors' and do not necessarily reflect those of the sponsors. We are thankful to the anonymous reviewers for their constructive comments

## References

Agrawal, R., Gollapudi, S., Kenthapadi, K., Srivastava, N., and Velu, R. (2010). Enriching textbooks through data mining. In *Proceedings of the First ACM Symposium on Computing for Development*, page 19. ACM.

Andersen, R., Chung, F., and Lang, K. (2006). Local graph partitioning using pagerank vectors. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pages 475–486. IEEE.

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. (2007). Dbpedia: A nucleus for a web of open data. In Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., and Cudr-Mauroux, P., editors, *The Semantic Web*, volume 4825 of *Lecture Notes in Computer Science*, pages 722–735. Springer Berlin / Heidelberg.

Baraniuk, R. (2008). *Challenges and opportunities for the open education movement: A Connexions case study*, pages 229–246. MIT Press, Cambridge, Massachusetts.

Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka Jr, E., and Mitchell, T. (2010). Toward an architecture for never-ending language learning. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, pages 1306–1313.

Carr, N. (2011). *The shallows: What the Internet is doing to our brains*. WW Norton & Co Inc.

Chen, S. and Magoulas, G. (2005). *Adaptable and adaptive hypermedia systems*. IRM Press.

Etzioni, O., Banko, M., and Cafarella, M. (2006). Machine reading. In *Proceedings of the National Conference on Artificial Intelligence*.

Fleiss, J. (1981). The measurement of interrater agreement. *Statistical methods for rates and proportions*, 2:212–236.

Kwok, C., Etzioni, O., and Weld, D. (2001). Scaling question answering to the web. *ACM Transactions on Information Systems (TOIS)*, 19(3):242–262.

Leskovec, J., Huttenlocher, D., and Kleinberg, J. (2010). Governance in social media: a case study of the wikipedia promotion process. In *AAAI International Conference on Weblogs and Social Media (ICWSM '10)*. AAAI.

Mihalcea, R. and Csomai, A. (2007). Wikify!: linking documents to encyclopedic knowledge. In *CIKM*, volume 7, pages 233–242.

Milne, D. and Witten, I. (2008). Learning to link with wikipedia. In *Proceeding of the 17th ACM conference on Information and knowledge management*, pages 509–518. ACM.

Noël, S. and Robert, J. (2004). Empirical study on collaborative writing: What do co-authors do, use, and like? *Computer Supported Cooperative Work (CSCW)*, 13(1):63–89.

Pavlik, P., Cen, H., Wu, L., and Koedinger, K. (2008). Using item-type performance to covariance to improve the skill acquisition of an existing tutor. In *Proc. of the 1st International Conference on Educational Data Mining*.

Pierrakos, D., Paliouras, G., Papatheodorou, C., and Spyropoulos, C. (2003). Web usage mining as a tool for

personalization: A survey. *User Modeling and User-Adapted Interaction*, 13(4):311–372.

Tong, H., Faloutsos, C., and Pan, J.-Y. (2006). Fast random walk with restart and its applications. In *Proceedings of the Sixth International Conference on Data Mining*, ICDM '06.

Vuong, A., Nixon, T., and Towle, B. (2011). A method for finding prerequisites within a curriculum. In *Proc. of the 4th International Conference on Educational Data Mining*.

Yates, A., Cafarella, M., Banko, M., Etzioni, O., Broadhead, M., and Soderland, S. (2007). Textrunner: Open information extraction on the web. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 25–26. Association for Computational Linguistics.

Yeh, E., Ramage, D., Manning, C., Agirre, E., and Soroa, A. (2009). Wikiwalk: random walks on wikipedia for semantic relatedness. In *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing*, pages 41–49. Association for Computational Linguistics.

Zheng, Q., Booth, K., and McGrenere, J. (2006). Co-authoring with structured annotations. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 131–140. ACM.