Using Prerequisites to Extract Concept Maps from Textbooks

Shuting Wang†, Alexander G. Ororbia II‡, Zhaohui Wu†, Kyle Williams‡,
Chen Liang‡, Bart Pursel*, C. Lee Giles‡†
†Computer Science and Engineering
‡Information Sciences and Technology
*Teaching and Learning with Technology
Pennsylvania State University, University Park, PA 16802, USA
sxw327@cse.psu.edu, ago109@ist.psu.edu,
{zzw109,kwilliams}@psu.edu, cul226@ist.psu.edu,bkp10@psu.edu,
giles@ist.psu.edu

ABSTRACT

We present a framework for constructing a specific type of knowledge graph, a *concept map* from textbooks. Using Wikipedia, we derive prerequisite relations among these concepts. A traditional approach for concept map extraction consists of two sub-problems: key concept extraction and concept relationship identification. Previous work for the most part had considered these two sub-problems independently. We propose a framework that jointly optimizes these sub-problems and investigates methods that identify concept relationships. Experiments on concept maps that are manually extracted in six educational areas (computer networks, macroeconomics, precalculus, databases, physics, and geometry) show that our model outperforms supervised learning baselines that solve the two sub-problems separately. Moreover, we observe that incorporating textbook information helps with concept map extraction.

Categories and Subject Descriptors

I.2.6 [**Learning**]: Knowledge acquisition; Concept learning; I.7.5 [**Document and Text Processing**]: Document Capture—*Document Analysis*; H.3.3 [**Information Storage And Retrieval**]: Information Search and Retrieval

Keywords

Open education; concept maps; textbooks; Web knowledge;

1. INTRODUCTION

A *knowledge graph* organizes knowledge by linking entities with their relationships and is applicable to many NLP tasks such as question answering [43] and knowledge acquisition [10]. While recent work has addressed reasoning in knowledge graphs (DBpedia [1] and YAGO [33]), and for real world facts [26], there has been little effort in organizing knowledge for educational purposes. Applications of such knowledge structures in education have been widely used in teaching and learning assessment [17].

There are many interesting challenges in extracting knowledge graphs for education. In some cases, nodes in an educational knowledge graph can be scientific and mathematical concepts, such as "Lasso" and "Regularization", instead of typical entities such as individuals, locations, or organizations. As such, instead of using general concept relationships such as "is-a" and "part-of", we focus on the *prerequisite* dependencies among concepts. A prerequisite dependency requires that learning one concept is necessary before learning the next. For instance, we need to have basic knowledge of "Regularization" in order to learn "Lasso" (or L1-regularized regression).

We present a method for constructing a specific type of knowledge graph, a **concept map**, which is widely used in the learning sciences [41]. In such a directed graph, each node is a scientific concept and directed links between these nodes imply their *prerequisite* dependencies. Figure 1 shows an example of an extracted concept map in the economics area where each node is an economical concept such as "Gross domestic product" and "Consumer price index" and links indicate prerequisite dependencies relating these concepts (from prerequisites to subsequents).

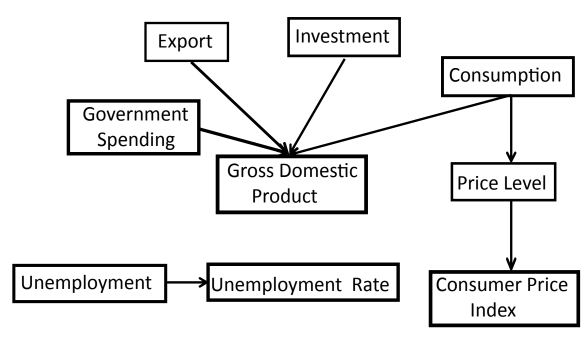


Figure 1: Example of an extracted concept map in economics.

CIKM'16 , October 24-28, 2016, Indianapolis, IN, USA

DOI: http://dx.doi.org/10.1145/2983323.2983725

Traditional approaches to knowledge graph extraction generally consist of two separate steps: **1)** Extracting key concepts and, **2)** Identifying relationships between key concepts. While these two common information extraction tasks have been well studied [7, 33, 2], solving these two tasks independently for educational content poses problems. We argue that these two problems are actually strongly coupled, meaning that the results of one affects the results of the other. Thus, solving these sub-problems independently might lead to sub-optimal performance. For example, in educational resources, a concept is often presented by first introducing its prerequisites. Thus the order in which two concepts appear in a document source can help identify their prerequisite relation. If a concept in this ordered chain is not correctly extracted, its prerequisite relation to other concepts will be lost. Furthermore, if this concept is the prerequisite to many others, we may no longer identify an important key concept.

Leveraging information from existing educational resources, we propose a concept map extraction model that jointly optimizes these two sub-problems. It then utilizes identified prerequisites to refine the extracted key concepts and vice versa. This model produces a related key concept set with prerequisite relations among those concepts.

There are many education resources from which one could build concept maps. For this work, we focus on textbooks since they often provide a comprehensive list of domain concepts and are often used as major educational resources in schools, colleges and universities. Educational resources such as textbooks and slides can provide implicit knowledge structures for knowledge graph extraction. For example, structural information such as table of contents (TOC) of a textbook can be very useful in identifying concept relationships. We feel this method could be easily generalized to other education resources with structured information such as slides and courses. We then augment "inside-the-book" knowledge with web content (for now, Wikipedia), thus enriching the content of a specific book with complementary information. As described in Section 4.4, we will empirically verify that using such complementary resources can give quality concept information at the secondary school and undergraduate level. In summary, our contributions are:

• The first attempt, to the best of our knowledge, to use textbooks to extract concept maps with explicit prerequisite relationships among the concepts.

• A set of principled methods that utilize both Web knowledge (Wikipedia) and the rich structure in textbooks to identify prerequisite relationships among domain concepts.

• An optimization framework that jointly solves the two sub-problems of concept map extraction and linkage.

• The generation of datasets from books in six different educational domains to show how our methods work.

Related work is introduced in Section 2. The joint optimization model for concept map extraction is presented in Section 3. We discuss the data preparation and baseline models in Section 4 and experimental results in Section 5. A case study on the subject of geometry is presented in Section 6 followed by conclusions and future work.

2. RELATED WORK

Early work on the problem of identifying knowledge graphs [6] inferred knowledge bases from a collection of noisy facts. More recently, ontology was used in the construction of a knowledge graph [13, 27]. [13] refined a knowledge base using relations and candidate facts found in an ontology. Building on earlier work [13], a probabilistic soft modeling framework [27] was used to jointly infer entities, categories and relations. Knowledge graph completion uses external sources (e.g., free text) to extract text patterns for certain relationships [3, 23, 29, 12, 34, 45, 8, 37]. In this line of work, relational facts are considered among existing word entities with a focus mainly on completing and extending an existing knowledge base. Our work differs from these work in that we consider scientific and mathematical concepts and the prerequisite dependencies between these concepts.

Other related research is key phrase detection using Wiki pedia. Early work [4] explored Wikipedia as a resource for detecting key phrases in open text and used a disambiguation statistical learning method to compare lexical context around an ambiguous named entity in the content of candidate Wikipedia pages. Previous work also identified key phrase by considering the interdependence between Wikipedia candidates [9, 11, 19, 22, 15, 28] to obtain coherent key phrases for documents.

More related we feel, concepts have been extracted from textbooks[38] and the textbook structure was used to organize the concepts, but this was done without considering explicit relationships between concepts. Instead of solely extracting entities from documents, our work constructs a concept map with both key concepts and their relationships. Moreover, our optimization model reinforces the mutual importance between the key concept extraction and prerequisite relationship identification and jointly optimizes the two sub-problems.

For concept prerequisite inference, [35] utilized PageRank and a random walk with restart scores [35]. The difference between reference links in any two Wikipedia articles [16] was also considered and a learning-to-rank method [44] constructed a concept graph with prerequisite relationships between courses.

Extracting concept maps with prerequisite relationships have also been studied in e-learning [36, 14, 39]. Concept maps [36] were derived from prerequisite relationships from ontologies by translating instances and interactions among instances into concept relationships in a knowledge graph. Association rule mining [14] was applied on learners' test records to derive concept maps with prerequisite relations. [39] explored prerequisite relationships among concepts by looking at topic coverage of each concept.

3. JOINT KNOWLEDGE GRAPH EXTRACTION FROM TEXTBOOKS

Here, we introduce our notation and describe how we jointly extract key concepts as well as the prerequisite relations. We define $c \in C$ as a concept where C is a set of Wikipedia concepts, $s \in S$ for a subchapter in the textbook. The term "subchapter" refers to all the headings in the TOC. For instance, both 1.1 and 1.1.1 are subchapters. A *key concept* in a book chapter is a concept which is not only mentioned but also discussed and studied in the subchapter. The input to our extractor consists of a *digital book* B with a list of titles, chapter number and contents for all its chapters. Each chapter contains one or more key concepts. The output is a *concept map* G, which is represented as a set of triples in the form $\{(c_1, c_2, r)|c_1, c_2 \in C, r \in R\}$, where $R = \{0, 1\}$ is the prerequisite relationship, r takes value 0 when c_1 and c_2 have no prerequisite relation; and takes value 1 when c_1 is c_2's prerequisite. We use $CS = \{cs_{ip} \in \{0, 1\}|1 \leq i \leq |C|, 1 \leq p \leq |S|\}$, to indicate concept appearance in subchapter, where cs_{ip} takes value 1 when the i^{th} concept is a key concept in the p^{th} subchapter; otherwise takes 0. Our goal is to optimize CS and R in order to obtain a global concept map.

3.1 Concept Map Extraction

3.1.1 Key Concept Extraction

Intuitively, if concept c is a key concept in subchapter s, it should have these few properties: **1). Local Relatedness:** Key concept c should be strongly related to subchapter s. For instance, the concept and book chapter share similar topics; **2). Global Coherence**: We argue that extracted key concepts should be coherent in the following way: **Less redundancy**: Chapters do not always discuss all of the same concepts. Information overlap between concepts in different chapters should be minimized. For instance, given a geometry textbook, if subchapter 2.1 covers "Triangle" in detail, subchapter 3.1 should not cover this concept in detail again.

Note that here we mention both concept-concept relatedness and concept-chapter relatedness. We denote all relatedness as one symmetric similarity function $f(\cdot, \cdot)$, which can take both the concept and chapter as arguments. We will discuss the definition of $f(\cdot, \cdot)$ later in this section. Given $f(\cdot, \cdot)$, the following objective function is proposed to derive the concept-subchapter matrix CS from the aforementioned properties:

$$P_1(CS) = \alpha_1 \sum_{i=1}^{|C|} \sum_{p=1}^{|S|} cs_{ip} f(c_i, s_p) \quad (1)$$

$$+\alpha_2 \sum_{i,j=1}^{|C|} \sum_{p \neq q}^{|S|} cs_{ip} cs_{jq} f(c_i, c_j), \quad (2)$$

where $I(\cdot) \in \{1, -1\}$ is an indicator function and returns 1 if the statement holds and returns -1 otherwise. αs are the term weights.

The first term corresponds to the *local relatedness* attributes and captures the relatedness between candidates and book chapter. This term should be maximized to select candidates similar to the book chapter. The second term is used to reduce redundancy in the concept map. For this term, we calculate the pairwise similarity between selected concepts in different chapters as the redundancy in the extracted concept map and this value should be minimized.

3.1.2 Prerequisite Relationship

We consider a pair of concepts to have a prerequisite relationship if they are: **3). Topically Related:** If two concepts cover different topics, it is unlikely that they have prerequisite relationships. **4). Complexity Level Difference:** Not all pairs of concepts with similar topics have prerequisite relationships. For example, "isosceles triangle" and "right angled triangle" cover similar topics but do not have learning dependencies. Thus, given two concepts, it is necessary to identify whether one concept is basic while another one is advanced. We denote complexity level of a concept as $l(\cdot)$ and later discuss the definition of $l(\cdot)$. Given $l(\cdot)$, we define the following optimization function for R:

$$P_2(R) = \alpha_3 \sum_{i,j=1, i \neq j}^{|C|} r_{ij} f(c_i, c_j) + \alpha_4 \sum_{i,j=1}^{|C|} r_{ij} (l(c_i) - l(c_j)). \quad (3)$$

The first term corresponds to the *Topically Related* attributes and should be maximized. The second term is used to measure the *Complexity Level Difference* between two concepts and we want this value to be maximized.

3.1.3 Joint Modeling

To reinforce the mutual benefit between two sub-problems, we propose **5). Order coherence**: Concepts should not be discussed without introducing their prerequisites, i.e., given a concept, prerequisite concepts should be introduced before this concept and subsequent concepts should be introduced after the concept. The following function is proposed to derive this mutual benefit property:

$$P_3(CS, R) = \alpha_5 \sum_{i,j=1}^{|C|} \sum_{p,q=1}^{|S|} I(p < q) cs_{ip} cs_{jq} r_{ij}. \quad (4)$$

In summary, the global objective function $\Lambda(CS, R) = P_1(CS) + P_2(R) + P_3(CS, R) + \beta_1 \|CS\| + \beta_2 \|R\|$ consists of P_1 for key concept extraction, P_2 for prerequisite relationship extraction, P_3 for mutual benefit modeling and L_1 regularization terms to control model complexity and is maximized.

3.1.4 Optimization

We maximize Λ to obtain the optimal concept map by adopting the Metropolis-Hasting algorithm to optimize CS and R respectively. $\forall cs \in CS$, we calculate the value of Λ using current value of cs and the flipped value of cs (denoted as cs'). We follow the following update rule to update CS.

$$\sigma_{CS}(cs, cs') = \begin{cases} 1, \ if \ \Lambda(R^{(n)}, CS^{(n)}, cs') \leq \Lambda(R^{(n)}, CS^{(n)}, cs). \\ e^{-\beta(\Lambda(R^{(n)}, CS^{(n)}, cs') - \Lambda(R^{(n)}, CS^{(n)}, cs))}, \ otherwise. \end{cases}$$

Similarly, $\forall r \in R$, we perform updates according to the following update rule:

$$\sigma_R(r, r') = \begin{cases} 1, \ if \ \Lambda(R^{(n)}, CS^{(n)}, r') \leq \Lambda(R^{(n)}, CS^{(n)}, r). \\ e^{-\beta(\Lambda(R^{(n)}, CS^{(n)}, r') - \Lambda(R^{(n)}, CS^{(n)}, r))}, \ otherwise. \end{cases}$$

3.2 Representation Schemes

We explore different schema for book chapter/concept content representation and then derive measures for concept/book chapter similarity $f(\cdot, \cdot)$ and the concept complexity level $l(\cdot)$. If multiple measures are derived for the same attribute, we adopt an equal weighted sum of different measures as the value of this attribute.

3.2.1 Word Based Similarity

We represent each chapter using words appearing in the chapter and each concept using a bag-of-word representation from the word content in their Wikipedia pages. Standard text preprocessing/weight procedures, including case-folding, stop-word removal and term frequency-inverse document frequency(TF-IDF) are applied. Based on this representation, we define the concept-chapter similarity function $f(\cdot,\cdot)$ (applied in Equation 1) as a combination of the following measures:

• *Title match*: This feature measures the relatedness between the concept tile and the chapter/concept title. Given a book chapter/concept title tb and a Wikipedia candidate title tw, if tw is in tb or tw is tb , $Titlematch(tb, tw) = 1$; Otherwise, $Titlematch(tb, tw) = 0$.

• *Content cosine similarity:* This feature measures the cosine similarity between the word TF-IDF vectors of chapter/concept contents.

• *Title Jaccard similarity:* This feature computes the Jaccard similarity between the chapter/concept title.

• *Sustained periods in subchapters:* A sustained period of a concept in a subchapter is the period from its first appearance to its last appearance. When the sustained period of a candidate concept in a subchapter is longer, it is more likely that this concept is important in this chapter.

We introduce one additional measure for concept-concept similarity. This concept-concept measure together with the other four aforementioned measures are used for concept-concept similarity and applied in Equation 2, and the first term in Equation 3.

• *Concept co-occurrences:* by counting the co-occurrences of two concepts lies within a sentence from either a book chapters or a Wikipedia page.

We also derive the following measures for a concept's complexity level based on its Wikipedia anchors. These measures are used in the second term in Equation 3.

• *Supportive relationship in concept definition:* A is likely to be B's prerequisite if A is used in B's definition. Here, we use the first sentence in the concept's Wikipedia page as its definition. $Supportive(A, B) = 1$ if A appears in B's definition. For instance, "Logarithm" is used to define "Natural logarithm" whose definition is "The natural logarithm of a number is its logarithm to the base e... " and $Supportive(logarithm\ , natural\ logarithm) = 1$.

3.2.2 Word Embeddings

This method maps concepts from the vocabulary to vectors of real numbers in a low-dimensional space [21]. We use word2vec which discovers lower dimensional vectors with two-layer neural networks using the contexts and syntactic of concepts. The concept similarity is defined as the *cosine similarity* of two concepts' embeddings, which is used in Equation 2 and the first term in Equation 3.

3.2.3 Wikipedia Anchors

Besides the content information, millions of cross-page links in Wikipedia are also useful in detecting concept relatedness and concept complexity levels. Given two concepts, we calculate the following measures as their similarity and use these measures in Equation 2, and the first term in Equation 3.

• *Wikipedia link based Jaccard similarity:* Given two concept, this feature computes the Jaccard similarity of the in-links/out-links of their Wikipedia page.

• *Wikipedia link based semantic similarity:* This feature computes the semantic relatedness of two concepts based on their Wikipedia links [42].

$$1 - \frac{\max(\log|Q_i|, \log|Q_j|) - \log|Q_i \cap Q_j|)}{\log W_{all} - \min(\log|Q_i|, \log|Q_j|)},$$

where Q_i is the set of Wikipedia concepts which link to w_i and W_{all} be the total number of concepts in Wikipedia.

We also derive the following measures for a Wikipedia concept's complexity level and use these three measures in the second term in Equation 3.

• *Number of in-links/out-links:* This feature returns the number of in-links/out-links in the concept Wikipedia page.

• *Relational strength in textbook/Wikipedia:* Relational strength $RS(w_i, w_j)$ measures the semantic relatedness between two concepts using concept co-occurrence and distance in the same sentence [5]:

$$log(\frac{n_{ij}/max(n)}{avg\_d_{ij}^2/max(avg\_d^2)}), i \neq j,$$

where n_{ij} is the co-occurrence of concept i and j within a sentence and $avg\_d_{ij}^2$ is sum of the distance of two keywords divided by the number of times two keywords appeared in the same sentence. If two concepts are close within in a lot of sentence in articles, implies that their relationship is also stronger than the others.

$$avg\_d_{ij}^2 = \frac{\sum_{m=1}^{n_{ij}} d_m^2}{n_{ij}}, i \neq j,$$

• *RefD:* [16] defines a new metrics measuring the prerequisite relationships between concepts using Wikipedia links between two concepts. If most related concepts of A refer to B but few related concepts of B refer to A, then B is more likely to be a prerequisite of A.

$$\frac{\sum_{i=1}^{|W|} v(w_i, B) \cdot u(w_i, A)}{\sum_{i=1}^{|W|} w(w_i, A)} - \frac{\sum_{i=1}^{|W|} v(w_i, A) \cdot u(w_i, B)}{\sum_{i=1}^{|W|} w(w_i, B)},$$

where $\mathcal{W} = \{w_1, ..., w_{|W|}\}$ is the concept space and $|W|$ is the size of the concept space; $u(w_i, A)$ weights the importance of w_i to A; and $v(w_i, A)$ is an indicator showing whether w_i has a Wikipedia link to A.

3.2.4 Textbook Structure

The TOC of textbooks contains implicit prerequisite relationships between concepts since textbooks usually introduce concepts based on their learning dependencies. Therefore, we define *TOC distance* between two concepts as the distance between their subchapter numbers. This feature is used to measure complexity level difference between concepts and applied in the second term in Equation 3.

Given two concepts A and B, a_i and b_i are used to denote their chapter number arrays. For example if A is in chapter 1.1, then $a_1 = 1$ and $a_2 = 1$. We define the *TOC distance* in textbooks between A and B as: $TOCdistance(a, b) = (a_i - b_i)/(\beta^{i-1})$ where i is the smallest index such as $a_i \neq b_i$ and β is a pre-specified decay parameter which is empirically set as 2. For instance, given a concept "HTTP" from chapter 2.3.1 and "HTTP message body" from chapter 2.3.2, TOC distance between them is 0.25 and "HTTP" could be "HTTP message body"'s prerequisite. Notice that a concept can serve as the key concept in multiple chapters and the value of the *TOC distance* feature between two concepts is the average TOC distance of all pairs of TOC of these two concepts. This measure is used in second term in the second term in Equation 3.

4. EXPERIMENT SETTINGS

4.1 Dataset

In order to build a test bed for concept map extraction, we manually construct concept maps using six widely-used textbooks: computer networking [1], macroeconomics [2], precalculus [3], databases [4], physics [5], and geometry [6].

To construct the final dataset, we first manually label key concepts: 1) Extract all Wikipedia concepts that appear in each book chapter. 2) Given a candidate concept c_i with title tw, we select it as a key candidate of subchapter j if $Titlematch(tw, tb_j) = 1$ where tb_j is the title of the subchapter j, or c_i is ranked within $top-30$ among all candidates in subchapter j based on *Content cosine similarity* feature. 3) Label the candidates as "key concept" or "not key concept" and obtain a set of key concepts for this area. Then for each pair of key concepts A and B , we manually label them as "A is B's prerequisite", "B is A's prerequisite" or "No prerequisite relationship". Table 1 shows characteristics of the dataset. For each area, three graduate students with corresponding background knowledge are recruited to label the data and we take a majority vote of the annotators to create final labels. We achieve an average 79% correlation for the key concept labeling task and an average 83% correlation for the concept relationship labeling task.

4.2 Baseline - Key Concept Extraction

4.2.1 *TextRank*

TextRank is a method widely used in key sentence and keyphrase extraction [20]. The general procedure of text rank is to build up a graph using candidate key concepts as vertices and co-occurrence of two candidates within a sentence as the weight on the edge between them. Then the algorithm iterates over the graph until it converges and sorts vertices based on their final scores to identify key concepts.

4.2.2 *Wikify*

Wikify detects significant Wikipedia concepts within unstructured texts. We use Wikipedia Miner developed in [22] to link book contents with Wikipedia concepts.

4.2.3 *Supervised Key Concept Extraction (*Supervised KCE*):*

Based on the local relatedness and global coherence attributes proposed in Section 3.2, we propose the following features for key concept learning from each subchapter.

[1]Kurose, James. F. (2005). Computer networking: a top-down approach featuring the Internet. Pearson Education India.

[2]Mankiw, N. Gregory.(2014). Principles of macroeconomics. Cengage Learning.

[3]Stewart, James, Lothar Redlin, and Saleem Watson. Precalculus: Mathematics for calculus. Cengage Learning, 2015.

[4]Ramakrishnan, Raghu, and Johannes Gehrke. "Database management systems.", 2000.

[5]Mark Horner, Samuel Halliday, Sarah Blyth, Rory Adams, Spencer Wheaton, "Textbooks for High School Students Studying the Sciences", 2008

[6]Dan Greenberg, Lori Jordan, Andrew Gloag, Victor Cifarelli, Jim Sconyers,Bill Zahnerm, "CK-12 Basic Geometry"

Local Features: We use features defined in Section 3.2 which capture the relatedness between concepts and book subchapters, i.e., *Title Match*, *Content cosine similarity*, *Title Jaccard similarity*, *Sustained periods in subchapters* of the concept, are used as local features in concept extraction.
Global Features: Global features include two sub-set of features: redundancy features and order coherence features.
Redundancy Features: This set of features measure the information overlap that a candidate c_i can possibly bring into the extracted concept set. Given the i^{th} candidate in j^{th} chapter, we calculate the similarity between this candidate and other candidates in different subchapters as the value of the redundancy feature of this candidate:

$$Red(cs_i) = \sum_{i,j=1}^{|C|} \sum_{p \neq q}^{|S|} cs_{ip} cs_{jq} f(c_i, c_j),$$

where $f(cs_{ki}, cs_{pj})$ is the similarity between candidate cs_{ki} and cs_{pj} and where $I(\cdot) \in \{1, -1\}$ is an indicator function and returns 1 if the statement holds and returns -1 otherwise.

Section 3.2 defines different semantic relatedness measurements and all these measurements can be applied to calculate redundancy features.
Order Coherence Features: Besides less redundancy attributes, we also expect consistent learning order in concepts extracted from the book, i.e., given a concept cs_{ki} in subchapter i, we expect that all cs_{ki}'s prerequisites appear in subchapters before i and all cs_{ki}'s subsequent concepts appear in subchapters after i. Given candidate cs_{ki} in the k^{th} subchapter, we define features *orderCorr* to capture the global learning order of the extracted concepts :

$$orderCorr(c_i) = \frac{\sum_{j=1}^{|C|} \sum_{p,q=1}^{|S|} I(p<q) cs_{ip} cs_{jq} r_{ij}}{\sum_{j=1}^{|C|} \sum_{p,q=1}^{|S|} cs_{ij} cs_{jq} |r_{ij}|}.$$

Equation 4.2.3 computes the percentage of concepts that are appropriately ordered based on the c_i's prerequisite relationships.

We use SVM^{rank} to predict rankings of Wikipedia candidates for each subchapter with data from one book as testing data and data from other five as training data.

4.3 Baseline - Prerequisite Relationship Identification

4.3.1 *Hyponym-Hypernym*

A hyponym is a concept whose semantic field is included within that of another concept (hypernym) and in this work, we use hyponym-Hypernym to as a baseline method of deriving prerequisite relationships. Lexico-syntax pattern based extraction methods are popular methods for extracting hyponym relationships between concepts because they offer effective text processing text. We adopt the10 lexico-syntactic patterns selected for hyponymy-hypernymy pattern matching in [40], as shown in Table 2.

4.3.2 *Supervised Relationship Identification (*Supervised RI*)*

For concept relationship extraction, we utilize *Topically Relatedness Features* and *Complexity Level Difference Fea-*

Domain	Network	Economics	Precalculus	Geometry	Database	Physics
# subchapter	98	37	23	48	90	152
# Key concepts per chapter	3.72	4.54	3.09	3.16	1.41	3.54
Candidate concepts per subchapter	70.53	84.16	70.21	55.89	49.17	68.94
# labeled pairs	1500	877	1171	1305	529	1517
# pairs with relationships	257	157	222	186	96	208

Table 1: Physical characteristics of books. *# labeled pairs* is the number of candidate concept pairs labeled as whether two concepts have prerequisite relationships. *# pairs with relationships* is the number of concept pairs with prerequisite relationships in all the labeled pairs.

NP2 such as NP1	NP1, one of NP2
such NP2 as NP1	NP1 (and|or) other NP2
NP1 is (a|an) NP2	NP2 consist(s) of NP1
NP2 includ(s|es|ing) NP1	NP2 (like|, specially) NP1
NP1 (is|are) called NP2	NP1 (in|belong to) NP2

Table 2: Extracted Lexico-syntax patterns. NP1 represents a subsequent Noun Phrase (NP) and NP2 represents a prerequisite Noun Phrase (NP).

tures introduced in Section 3.2 to identify concept prerequisite relationship. *Topically Relatedness* measures include *Title match*, *Content cosine similarity*, *Title Jaccard similarity*, *Wikipedia link based Jaccard similarity*, *Wikipedia link based semantic similarity*, *Relational strength in textbook/Wikipedia*. *Complexity Level Difference* features include *Supportive relationship in concept definition*, *RefD*,*Number of in-links/out-links*, *TOC Distance*.

Then we perform a binary class classification using SVM to identify prerequisite relationships with five books as training data and one book as testing data.

4.4 Wikipedia Coverage of Concepts

Wikipedia has previously been utilized as a controlled vocabulary for topic indexing [18, 19] and key phrase extraction [24]. A few studies have examined Wikipedia coverage of academically related topics [25, 30, 31, 32]. Though some work showed that Wikipedia does not properly cover academic content on the front end of science, previous studies [25, 32] have demonstrated that Wikipedia's coverage of topics is comprehensive for secondary school and undergraduate education.

In order to further validate the coverage of the extracted concept maps, we conducted the following experiments. For each book, three graduate students with corresponding background knowledge are recruited to manually extract all concepts from each subchapter (randomly sampled from the book), and label whether these concepts have a corresponding Wikipedia page. We found that 88% of the concepts (Computer network: 85%, Macroeconomics: 86%, Precalculus: 91% , Geometry: 97%, Physics: 85%, Database: 89%) in the books are covered by Wikipedia and this provides some empirical evidence of reasonable coverage of the extracted concept maps.

4.5 Parameter Selection

As shown in Equation 2 and Equation 3, our concept maps are shaped by parameters $\alpha = \{\alpha_i, i = 1, 2, 3, 4\}$ and $\beta = \{\beta_j, j = 1, 2\}$ where αs are the term weight in the optimization function and βs are the weight of L1-regularization. We test different methods in a "leave one book out" manner, i.e, when testing on one book, we train our model using the other 5 books to select the optimal combination of parameters.

4.6 Model Initialization

• *Concept-Subchapter Matrix Initialization:* To initialize $CS(\cdot)$, we use two features *Title match* and *Content cosine similarity* proposed in Section 3.2 which measure the local similarity between a candidate and a book chapter. We set $cs_{ij} = 1$, i.e., candidate c_i is a key concept in subchapter j, if $Titlematch(c_i, tb_j) = 1$ where tb_j is the title of the subchapter j, or c_i is ranked within $top - 5$ based on cosine similarity between chapter/concept contents feature.
• *Concept Relationship Matrix Initialization:* To initialize the concept relationship matrix $R(\cdot)$, given two concepts c_i and c_j, we set $r_{ij} = 1$ if their complexity level difference is higher than threshold t_1 and topically relatedness is higher than threshold t_2. Empirically, t_1 is set as mean value of the overall complexity level difference and t_2 as mean value of the overall topically relatedness.

5. EXPERIMENTAL RESULTS

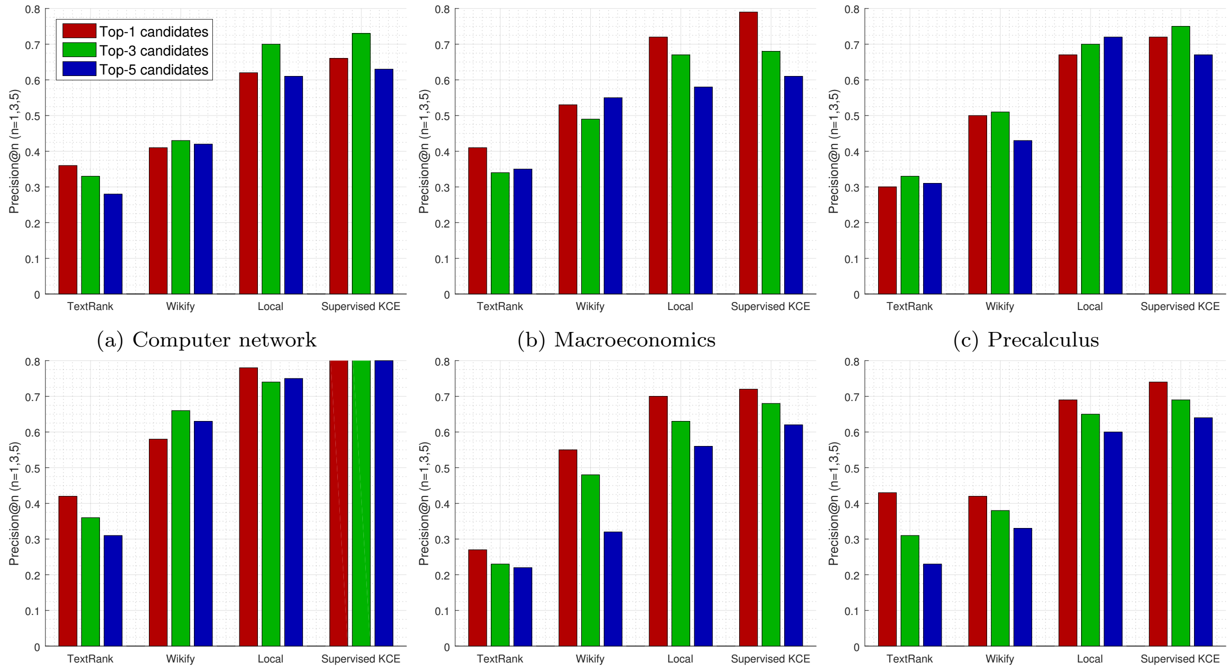
5.1 Effect of Textbook Information

In this section, we present how textbook structures help concept map extraction.

Figure 2 shows ranking precisions of key concept extraction on six books. For the baseline methods presented, we needed to manually decide the number of key concepts in each subchapter. We thus present the performance of $top-1$, $top-3$, and $top-5$ candidates from the concept extraction phase respectively. As shown, we test different combinations of features, with the local features derived from different aspects of relatedness between book subchapter and Wikipedia candidates, and global features which consider the global coherence of the book structure. The results show that incorporating our proposed global features (See "Supervised KCE" in Figure 2) into the extractor does achieve significantly higher precision than other methods which do not consider book structure (TextRank, Wikify and Local features).

In Table 3, we present the F-1 score of concept relationship identification using $top-1$, $top-3$, and $top-5$ candidates from the concept extraction phase respectively.

The results show that both features derived from Wikipedia and textbooks features achieve significantly higher F-1 score than hyponym-hypernym pattern does. Moreover, we observe that textbook features outperform Wikipedia features



(a) Computer network (b) Macroeconomics (c) Precalculus

(d) Geometry (e) Database (f) Physics

Figure 2: Precision@n (n=1,3,5) for key concept extraction from six textbooks. *Local* refers to the supervised learning model using local features defined in Section 4.2.3 with same experiment settings as *Supervised KCE*.

	Hyponym			Wiki			Textbook		
# candidate	1	3	5	1	3	5	1	3	5
Network	0.21	0.32	0.19	0.45	0.56*	0.54	0.49	**0.6***	0.57
Economics	0.25	0.36	0.3	0.55	0.56	0.5	0.55	**0.58**	0.52
Precalculus	0.29	0.42	0.36	0.52	0.56	0.47	0.44	0.52	**0.57**
Geometry	0.28	0.36	0.41	0.48	0.56*	0.53	0.55	**0.61***	0.59
Database	0.17	0.38	0.44	0.49	**0.55**	0.45	0.5	0.53	0.52
Physics	0.23	0.37	0.44	0.5	0.55	0.58*	0.6	0.58	**0.62***

Table 3: F-1 score for concept relationship prediction. *Hyponym* refers to the hyponym-hypernym baseline method. *Textbook/Wikipedia* are supervised learning methods with features textbooks/Wikipedia features using same experiment settings as *Supervised RI*. $*$ indicates when textbook features are statistically significantly better ($p < 0.01$) than the Wikipedia features.

(Textbook features include *Concept co-occurrence* in book chapters, *Relational strength* in book contents and *TOC distance* measures. Wikipedia features include *concept co-occurrence* in Wiki pages, *Content cosine similarity*, *RefD*, *Wikipedia link based semantic similarity*, *Relational strength* in Wikipedia and *Supportive relationship in concept definition* measures.). This is because textbooks are designed for education purpose and provides a better knowledge structure than web knowledge base does. Another potential reason is that features that utilize TOC information (*TOC distance* feature) consider not only concept complexity level difference but also their semantic relatedness. If two concepts are introduced in neighbor subchapters (say subchapter 1.1 and 1.2), this feature reveals that two concepts might be semantic similar between authors usually put related concepts in neighbor chapters.

5.2 Joint Optimization

Table 3 shows the prediction accuracy of the baseline methods and the joint optimization model.

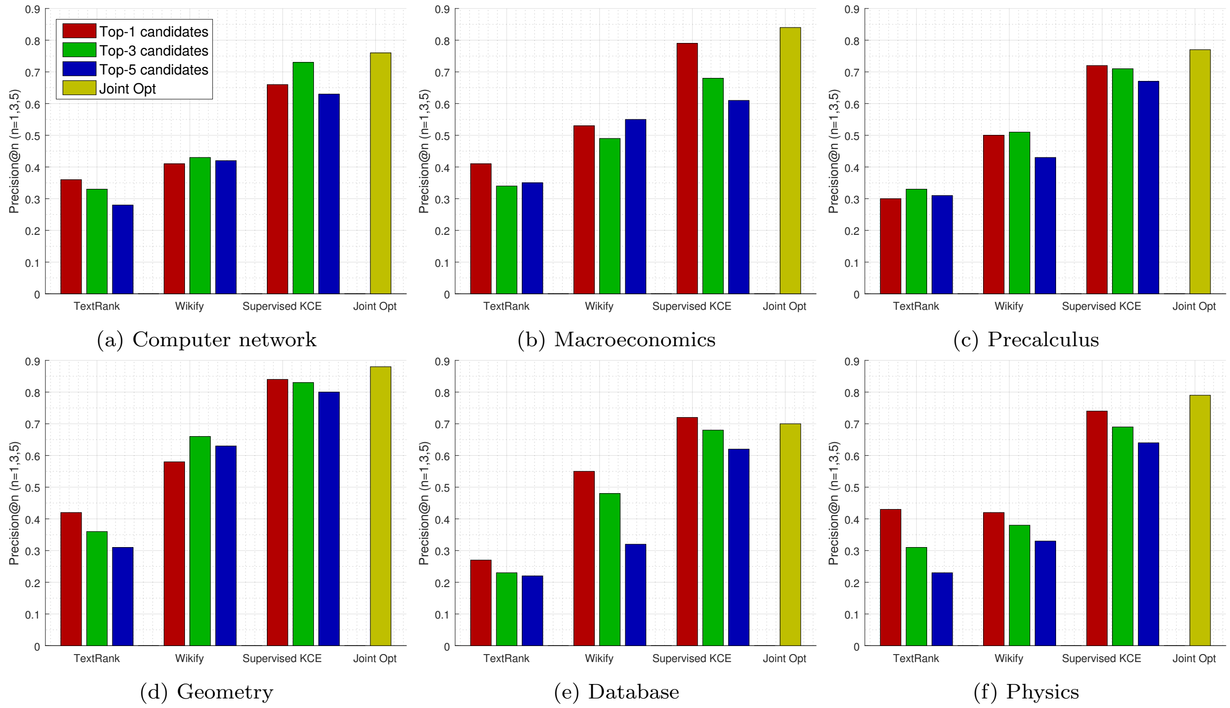
The proposed optimization model often outperforms all others with only an exception on precision@1 of database, as a trade-off to performance of concept relationship prediction, as shown in Table 4. Our joint optimization model consistently outperforms the strongly baseline in F1-score on all the six textbooks. In addition, the proposed model can decide the number of concepts in each subchapter automatically by optimizing the proposed objective function while the baseline models depend on a manually decided value.

	Supervised RI			Joint Opt
# candidate	1	3	5	
Network	0.52	0.62	0.57	**0.63**
Macroeconomics	0.55	0.67*	0.63	**0.7***
Precalculus	0.54	0.61	0.63*	**0.66***
Geometry	0.52	0.64	0.67*	**0.71**
Database	0.51	**0.58**	0.49	0.55
Physics	0.58	0.65*	0.62	**0.69***

Table 4: F-1 score for concept relationship prediction with/without joint optimization. $*$ indicates when the joint optimization model is statistically significantly ($p < 0.01$) better.

5.3 Measurement Importance

In this section, we develop some insights regarding feature importance by reporting performance of the concept extractor and relationship identifier using different feature combinations.



(a) Computer network (b) Macroeconomics (c) Precalculus

(d) Geometry (e) Database (f) Physics

Figure 3: Precision@n (n=1,3,5) for key concept extraction from six textbooks with/without joint optimization.

	Title Features			Content Similarity Features		
# candidate	1	3	5	1	3	5
Network	0.32	0.29	0.19	0.61	0.64	**0.65**
Macroeconomics	0.43	0.39	0.32	**0.72**	0.65	0.54
Precalculus	0.41	0.36	0.27	0.66	**0.68**	0.59
Geometry	0.46	0.38	0.32	**0.79**	0.76	0.7
Database	0.24	0.18	0.11	0.66	**0.69**	0.58
Physics	0.3	0.22	0.16	**0.68**	0.62	0.59

Table 5: Precision@n (n=1,3,5) for key concept extraction from six textbooks using different feature combinations. *Title/Content* features include local and global features derived from title/content information as defined in Section 4.2.3.

Table 5 shows the ranking precisions of measurements using title information and that using content information. As shown, content similarity features outperform title features since compared to subchapter titles, subchapter contents contain more information and achieve much higher recall rate in key concept extraction.

Figure 6 shows the performance of relationship prediction using *Topically Relatedness* features and *Complexity Level Difference* features defined in Section 4.3.2 respectively. We observe that complexity level difference features perform better than content similarity features. This suggests that by capturing which feature is more basic, we can achieve better performance than only considering topic relatedness between features. We also observe that more fundamental subjects such as precalculus, geometry and physics have better performance than advanced subjects such as computer networks and database. A potential reason is that for those domains, different textbooks provide very similar learning and TOC structures, while for advanced subjects

	Topically Relatedness			Concept Complexity Level		
# candidate	1	3	5	1	3	5
Network	0.41	0.49	0.46	0.56	**0.58**	0.5
Macroeconomics	0.47	0.52	0.43	0.55	**0.6**	0.51
Precalculus	0.53	0.58	0.61	0.55	0.6	**0.62**
Geometry	0.42	0.45	0.53	0.49	0.53	**0.56**
Database	0.52	0.59	0.56	0.53	0.59	**0.61**
Physics	0.3	0.22	0.16	**0.64**	0.62	0.59

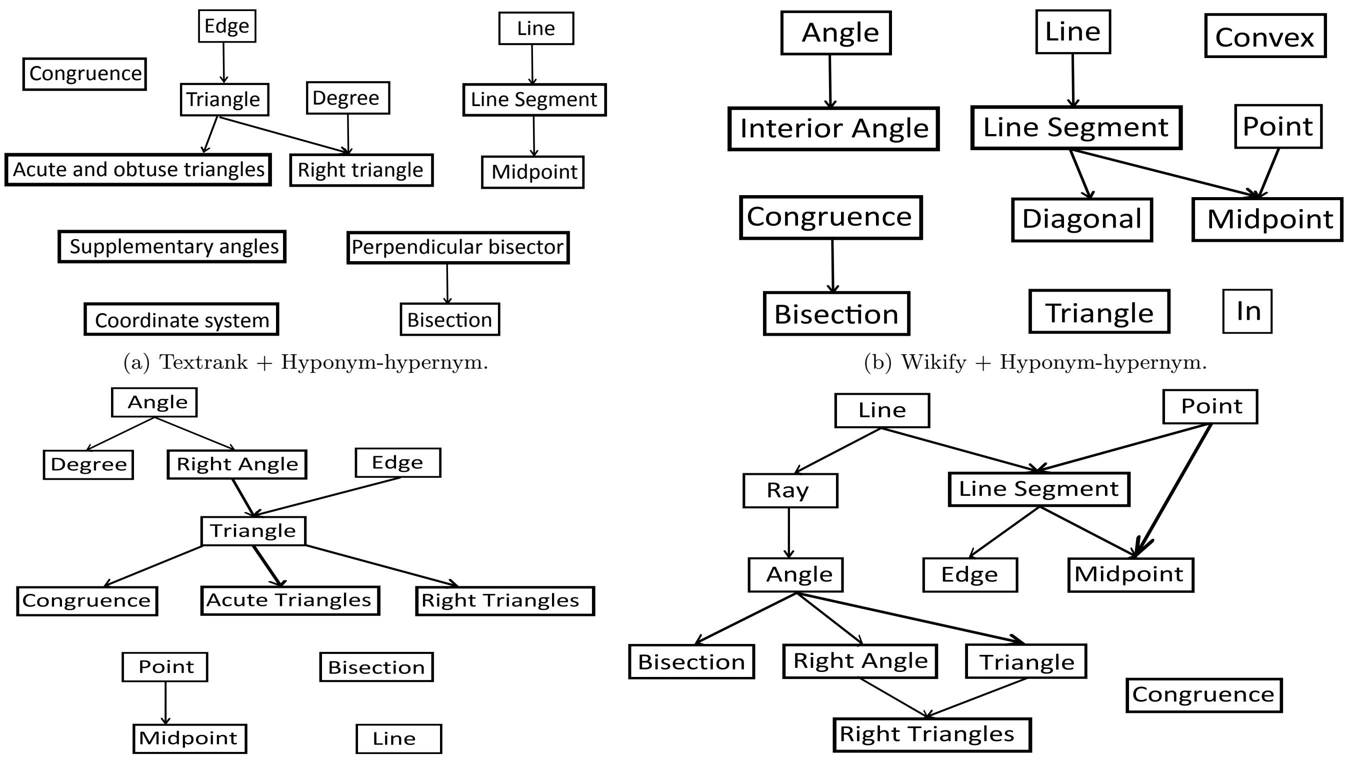
Table 6: F1-score@n (n=1,3,5) for the relationship prediction using different measures.

textbooks organize knowledge quite differently. However, this remains to be determined.

6. CASE STUDY

Here we present a case study on concept maps extracted for geometry. From Figure 4c and 4d, we can observe that by considering both the lexical similarity and semantic relatedness, the *Supervised KCE + Supervised RI* method, here and after, the supervised learning method, and *joint optimization* model achieve better performance in both concept extraction from book chapters and relationship identification than the *TextRank+Hyponym-hypernym* method (see Figure 4a) and *Wikify+Hyponym-hypernym* method (see Figure 4b). Moreover, supervised method and joint optmization model consider book structure information and this reaffirm the effectiveness of textbook structure in key concept extraction and concept relationship identification.

By capturing the mutual dependencies between two subproblems, the joint optimization model achieves better prediction accuracy than the supervised learning method. For instance, supervised learning method fails to extract "Ray" and the prerequisite dependency between "Ray" and "Angle"



(a) Textrank + Hyponym-hypernym.

(b) Wikify + Hyponym-hypernym.

(c) Supervised KCE + Supervised RI.

(d) Joint optimization model.

Figure 4: Case study for geometry concept maps.

while joint optimization model makes the correct prediction. A possible reason is that "Ray" is not extracted as a key concept because its content is not very similar to the content of the book chapter (compared to other candidates in this chapter). However, the joint optimization model identifies that "Ray" is likely to be prerequisite of "Angle" which is highly ranked as a key concept in some book chapters.

7. CONCLUSION AND FUTURE WORK

We describe measures that identify prerequisite relationships between concepts. We propose a joint optimization model for concept map extraction from textbooks that utilizes the mutual interdependency between key concept extraction and the identification of related concepts from knowledge structures in Wikipedia. Experiments on six concept maps created manually from six different textbooks show that the proposed method gives promising results. To our knowledge, this is the first work that utilizes the implicit prerequisite relationships embedded in textbook table of contents (TOCs) for prerequisite relationship extraction. Future directions would be to construct concept maps from multiple books from the same area or use similar textbooks to modify each others concept maps. Another direction would be to develop a semi-automatic method for building large scale education area concept maps.

8. REFERENCES

[1] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. *Dbpedia: A nucleus for a web of open data*. Springer, 2007.

[2] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction for the web. In *IJCAI*, volume 7, pages 2670–2676, 2007.

[3] A. Bordes, J. Weston, R. Collobert, and Y. Bengio. Learning structured embeddings of knowledge bases. In *Conference on Artificial Intelligence*, 2011.

[4] R. C. Bunescu and M. Pasca. Using encyclopedic knowledge for named entity disambiguation. In *EACL*, pages 9–16, 2006.

[5] N.-S. Chen, C.-W. Wei, H.-J. Chen, et al. Mining e-learning domain concept map from academic articles. *Computers & Education*, 50(3):1009–1021, 2008.

[6] W. W. Cohen, H. Kautz, and D. McAllester. Hardening soft information sources. In *SIGKDD*, pages 255–259. ACM, 2000.

[7] O. Etzioni, M. Cafarella, D. Downey, S. Kok, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Web-scale information extraction in knowitall:(preliminary results). In *WWW*, pages 100–110. ACM, 2004.

[8] M. Fan, D. Zhao, Q. Zhou, Z. Liu, T. F. Zheng, and E. Y. Chang. Distant supervision for relation extraction with matrix completion. In *ACL*, pages 839–849, 2014.

[9] P. Ferragina and U. Scaiella. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). pages 1625–1628, 2010.

[10] S. E. Gordon, K. A. Schmierer, and R. T. Gill. Conceptual graph analysis: Knowledge acquisition for

instructional system design. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 35(3):459–481, 1993.

[11] X. Han and J. Zhao. Named entity disambiguation by leveraging wikipedia semantic knowledge. In *CIKM*, pages 215–224. ACM, 2009.

[12] R. Hoffmann, C. Zhang, X. Ling, L. Zettlemoyer, and D. S. Weld. Knowledge-based weak supervision for information extraction of overlapping relations. In *ACL*, pages 541–550, 2011.

[13] S. Jiang, D. Lowd, and D. Dou. Learning to refine an automatically extracted knowledge base using markov logic. In *ICDM)*, pages 912–917. IEEE, 2012.

[14] R. Kavitha, A. Vijaya, and D. Saraswathi. An augmented prerequisite concept relation map design to improve adaptivity in e-learning. In *PRIME*, pages 8–13. IEEE, 2012.

[15] S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti. Collective annotation of wikipedia entities in web text. In *SIGKDD*, pages 457–466. ACM, 2009.

[16] C. Liang, Z. Wu, W. Huang, and C. L. Giles. Measuring prerequisite relations among concepts. 2015.

[17] J. R. McClure, B. Sonak, and H. K. Suen. Concept map assessment of classroom learning: Reliability, validity, and logistical practicality. *Journal of research in science teaching*, 36(4):475–492, 1999.

[18] O. Medelyan, E. Frank, and I. H. Witten. Human-competitive tagging using automatic keyphrase extraction. In *EMNLP*, pages 1318–1327, 2009.

[19] O. Medelyan, I. H. Witten, and D. Milne. Topic indexing with wikipedia. In *AAAI*, pages 19–24, 2008.

[20] R. Mihalcea and P. Tarau. Textrank: Bringing order into texts. In *ACL*, 2004.

[21] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[22] D. Milne and I. H. Witten. Learning to link with wikipedia. In *CIKM*, pages 509–518. ACM, 2008.

[23] M. Mintz, S. Bills, R. Snow, and D. Jurafsky. Distant supervision for relation extraction without labeled data. In *ACL*, pages 1003–1011, 2009.

[24] C. Q. Nguyen and T. T. Phan. An ontology-based approach for key phrase extraction. In *ACL-IJCNLP*, pages 181–184, 2009.

[25] C. Okoli, M. Mehdi, M. Mesgari, F. Å. Nielsen, and A. Lanamäki. Wikipedia in the eyes of its beholders: A systematic review of scholarly research on wikipedia readers and readership. *JASIST*, 65(12):2381–2403, 2014.

[26] J. Pujara, H. Miao, L. Getoor, and W. Cohen. Knowledge graph identification. In *ISWC*, pages 542–557. 2013.

[27] J. Pujara, H. Miao, L. Getoor, and W. Cohen. Large-scale knowledge graph identification using psl. In *AAAI*, 2013.

[28] L. Ratinov, D. Roth, D. Downey, and M. Anderson. Local and global algorithms for disambiguation to wikipedia. pages 1375–1384, 2011.

[29] S. Riedel, L. Yao, and A. McCallum. Modeling relations and their mentions without labeled text. In *ECML PKDD*, pages 148–163. Springer, 2010.

[30] E. K. Rush and S. J. Tracy. Wikipedia as public scholarship: Communicating our impact online. *Journal of Applied Communication Research*, 38(3):309–315, 2010.

[31] A. Samoilenko and T. Yasseri. The distorted mirror of wikipedia: a quantitative analysis of wikipedia coverage of academics. *EPJ Data Science*, 3(1):1–11, 2014.

[32] N. J. Schweitzer. Wikipedia and psychology: Coverage of concepts and its use by undergraduate students. *Teaching of Psychology*, 35(2):81–85, 2008.

[33] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: a core of semantic knowledge. In *WWW*, pages 697–706. ACM, 2007.

[34] M. Surdeanu, J. Tibshirani, R. Nallapati, and C. D. Manning. Multi-instance multi-label learning for relation extraction. In *EMNLP*, pages 455–465, 2012.

[35] P. P. Talukdar and W. W. Cohen. Crowdsourced comprehension: predicting prerequisite structure in wikipedia. In *Building Educational Applications Using NLP*, pages 307–315. Association for Computational Linguistics, 2012.

[36] S.-S. Tseng, P.-C. Sue, J.-M. Su, J.-F. Weng, and W.-N. Tsai. A new approach for constructing the concept map. *Computers & Education*, 49(3):691–707, 2007.

[37] Q. Wang, B. Wang, and L. Guo. Knowledge base completion using embeddings and rules.

[38] S. Wang, C. Liang, Z. Wu, K. Williams, B. Pursel, B. Brautigam, S. Saul, H. Williams, K. Bowen, and C. Giles. Concept hierarchy extraction from textbooks. In *DocEng*, 2015.

[39] S. Wang and L. Liu. Prerequisite concept maps extraction for automaticassessment. In *WWW*, pages 519–521, 2016.

[40] B. Wei, J. Liu, J. Ma, Q. Zheng, W. Zhang, and B. Feng. Motif-re: motif-based hypernym/hyponym relation extraction from wikipedia links. In *NIPS*, pages 610–619. Springer, 2012.

[41] R. T. White. *Learning science.* Basil Blackwell, 1988.

[42] I. Witten and D. Milne. An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *AAAI*, pages 25–30, 2008.

[43] S. Yang, F. Han, Y. Wu, and X. Yan. Fast top-k search in knowledge graphs. In *ICDE*, 2016.

[44] Y. Yang, H. Liu, J. Carbonell, and W. Ma. Concept graph learning from educational data. In *WSDM*, pages 159–168, 2015.

[45] X. Zhang, J. Zhang, J. Zeng, J. Yan, Z. Chen, and Z. Sui. Towards accurate distant supervision for relational facts extraction. In *ACL*, pages 810–815, 2013.