Automatic Concept Extraction for Intelligent Textbooks

Hung Chau
hkc6@pitt.edu

Abstract

The increasing popularity of digital textbooks as a new learning media resulted in a growing interest in developing a new generation of *intelligent textbooks* that can help readers to learn better, adapting to their learning goals and current state of knowledge. These intelligent textbooks are most frequently powered by internal knowledge models, which associate a list of unique domain knowledge concepts with each section of the textbook. With this kind of concept-level knowledge representations, a number of intelligent operations could be performed; e.g., student modeling, content linking or content recommendation. However, manual indexing of each textbook section with concepts is challenging, time-consuming and inconsistent. Automatic keyphrase extraction methods have been being developed over the last twenty years; however, few of the known approaches have been focusing on textbooks and evaluated on a full-scale textbook corpus. In this paper, we present a supervised machine learning method with a list of rich, carefully hand-crafted features. We evaluate our proposed approach using several state-of-the-art keyphrase extraction methods as a baseline on a newly constructed full-scale dataset. The results show that the proposed model outperforms all the baseline methods.

1 Introduction

More and more educational content today is found on the web – some of it on educational platforms such as Coursera or EdX, and a growing number of it in online textbooks, personal websites and blogs (a simple Google search for "LSTM tutorial", for example, returns 320,000 results). This content is diverse in its intended audience. Some of it may assume deep expertise in the subject; some may be introductory for people new to the field. Some may be aimed towards people in finance, and other for people in the medical field. When presented as part of a search result ranking, these textbook chapters, lessons and tutorials, lack the necessary context, such as the required prerequisites needed to grasp the material presented in them. To address this problem, recent work (Labutov and Lipson, 2016; Yang et al., 2015) had started to look towards automatically creating learning paths, complete with necessary prerequisites, from such unstructured educational content on the web. A

key input to such systems is the domain-specific terminology extracted from these educational documents. By understanding what domain-specific keywords (e.g., "gradient descent" in an article about "LSTMs") that are presented in the educational documents, and their distribution, these documents can then be "connected" together into sequences that progress from simpler to the more complex, and ultimately towards the learner's educational goal.

In this work, we focus on this fundamental task of *concept keyword extraction from educational texts*. Terminology, or keywords are used as proxies for the underlying concepts present in the documents, and their distribution in turn acts as a proxy for the degree to which these concepts are either assumed or explained in a given document. Both precision and recall at the task of concept keyword extraction are therefore critical for this downstream task of connecting educational content based on the degree to which one document is a prerequisite for another. This paper presents a thorough and systematic analysis of supervised learning applied to the task of concept extraction from educational content. Previous work at this task had either focused on non-educational keyword extraction, or had not analyzed and experimented with concept-extraction in educational context to the same level of coverage and depth as we do in this paper. The key contributions of this work are as follows:

- **Concept annotation**: We perform a rigorous and systematic annotation of concept keywords in a technical textbook. Improving our annotation protocol over multiple iterations, we achieve a relatively high inter-annotator agreement and ensure that the annotated keywords closely align to the underlying concepts.
- **Concept extraction**: We engineer and experiment with a highly encompassing feature set for learning to extract the annotated concepts. Our feature set spans both linguistic features and features encoding relative corpus statistics (i.e., summarizing relative word frequencies between technical and non-technical corpora).
- **Evaluation**: We perform systematic ablation studies of the proposed supervised model, as well perform extensive comparative evaluation with a number of keyword-extraction models proposed in literature.

2 Related Work

Automatic keyphrase extraction has been extensively studied and examined using different approaches such as rule-based, supervised learning, unsupervised learning or deep neural networks. Typically, automatic keyphrase extraction systems consist of two parts (Augenstein et al., 2017): (1) preprocessing data and extracting a list of *candidate keyphrases* using lexical patterns and heuristics; and then (2) determining which of these candidates are correct keyphrases based on some ranking scores.

The goal of extracting the candidate keyphrase list is to obtain all potential candidates while keeping the number of candidates as small as possible. Several studies extract candidates from words with certain part-of-speech (POS) tags (e.g., *nouns* or *noun-nouns*) (Mihalcea and Tarau, 2004; Bougouin et al., 2013; Liu et al., 2009a; Wan and Xiao, 2008). Others extract n-grams with simple filtering rules (Witten et al., 1999; Medelyan et al., 2009) or only allow those matching Wikipedia article titles (Wang et al., 2015; Grineva et al., 2009). More complex approaches extract noun phrases and apply predefined lexico-syntatic patterns (Florescu and Caragea, 2017; Le et al., 2016).

The next step is to score each candidate based on some properties that indicate how likely that candidate is a keyphrase in the given document. Machine learning approaches to this task can be grouped into those that are supervised or unsupervised. Among *unsupervised learning approaches*, *graph-based approaches* (Mihalcea and Tarau, 2004; Bougouin et al., 2013) consider a candidate keyphrase as important if it is related to a large number of candidates and those candidates are also important in the document. Candidates and the relations between them form a graph for the input document. A graph-based ranking (e.g., *PageRank*) is applied to give a score to each node. Finally, the top-ranked candidates are selected as keyphrases for the input document. Unsupervised *topic-based clustering methods* (Liu et al., 2009b, 2010; Grineva et al., 2009) attempt to group semantically similar candidates in a document as *topics*. Keyphrases are then selected based on the centroid of each cluster or the importance of each topic.

The *supervised learning approaches* typically frame this task as a *binary classification problem* (Witten et al., 1999; Hulth, 2003; Jiang et al., 2009). A variety of features have been used for training supervised models including *statistics*-based features *title*-based features, *linguistics*-based features or *external resources* (Hammouda et al., 2005; Witten et al., 1999; Rose et al., 2010; Hulth, 2003; Wang et al., 2015; Yih et al., 2006; Nguyen and Kan, 2007).

Deep learning approaches sharing features of both supervised and unsupervised learning, have been successfully applied to many NPL-related tasks including named entity recognition (NER) and sequence tagging. However, few studies focused on keyphrase extraction problem. (Meng et al., 2017) built a deep keyphrase generation with an encoder-decoder framework. They applied an RNN-based generative model to predict keyphrases.

While many general concept-extraction approaches exists, few focused on an educational domain and almost none on a textbook corpus. There are a number of projects that *apply* book concepts to achieve a specific target; for example, building concept hierarchies for textbooks (Wang et al., 2015) or separating prerequisite and outcome concepts (Labutov et al., 2017). However, they do not focus on the advanced concept *extraction* and use existing data (Labutov et al., 2017) or lightweight extraction approaches (Wang et al., 2015).

The work presented in this paper applies the state-of-the art extraction approaches to the under-explored textbook context. We use a supervised method for concept extraction from textbooks with an extensive list of carefully selected

features. We evaluate the approach on a brand new dataset and compare it with several state-of-art baselines. We made the code and data available on Github [1].

3 The Dataset

One of the challenges of keyphrase extraction is obtaining a good dataset for training and testing models. Especially, there are very few datasets with labeled data for educational resources such as textbooks, course descriptions, slides, e.t.c. An added challenge for the educational context is its focus on knowledge transfer. As a result, educational applications usually refer to *concepts* associated with text rather than keywords or keyphrases. In this context, we define *domain concepts* as keyphrases (single words or short phrases of two to four words) that represents most essential knowledge elements presented in a text fragment (e.g., a sentence, a paragraph, a section) in respect to its target domain (e.g., Computer Science (CS)) or related domain (e.g. Statistics). Those concepts should have specific meanings in the CS domain and be important in Information Retrieval (IR) sub-domain, but may have different meanings in other domains. Without understanding the conceptual meaning, readers could not understand the content. For example, considering the sentences/paragraphs below:

*"**Tokenization** is the task of chopping it up into pieces, called **tokens**, perhaps at the same time throwing away certain characters, such as punctuation."* In this example, ***Tokenization*** and ***tokens*** are domain concepts, but ***characters*** and ***punctuation*** are not.

To support our work on automatic concept extraction, we built a dataset with a section-level concept index for the first 16 chapters of Introduction to Information Retrieval (IIR) textbook[2]. For each section (the lowest-level TOC unit) of the textbook, the dataset provides a list of essential concepts mentioned in the section. The statistics of the dataset are shown in Table 1.

To build this dataset, we engaged three paid experts - one PhD student working in the IR domain and two Masters students who completed the IR course with high scores. Before the start of the process, the annotators received training and passed a test focused on the understanding of the task, the "codebook" of annotation rules, and the annotation interface. Every week three experts focused on completing annotations for one chapter (i.e., all sections belonging to the chapter). After finishing an annotation session, they discussed the cases in which their annotation disagreed, made the final decision for the concept list, and, if necessary, added new "codebook" rules to help increase the agreement in the future. Throughout this process, the inter-annotator proportion agreement among the three annotators *before discussion* had gradually increased from 0.25 to 0.68 at week 3 and 0.9 at the end (see Figure 1).

[1] https://github.com/ANONYMOUS/Concept-Extraction/

[2] https://nlp.stanford.edu/IR-book/

Characteristic	
Number of chapters	16
Number of sections	86
Number of all concepts	3175
Number of 1-grams	1121 (35.31%)
Number of 2-grams	1565 (49.29%)
Number of 3-grams	422 (13.29%)
Number of 4-grams	58 (1.83%)
Number of 5+6-grams	9 (0.28%)
Number of all unique concepts	1543
Number of unique 1-grams	278 (18.02%)
Number of unique 2-grams	871 (56.45%)
Number of unique 3-grams	330 (21.39%)
Number of unique 4-grams	55 (3.56%)
Number of unique 5+6-grams	9 (0.58%)

Table 1: Statistics of the IIR dataset

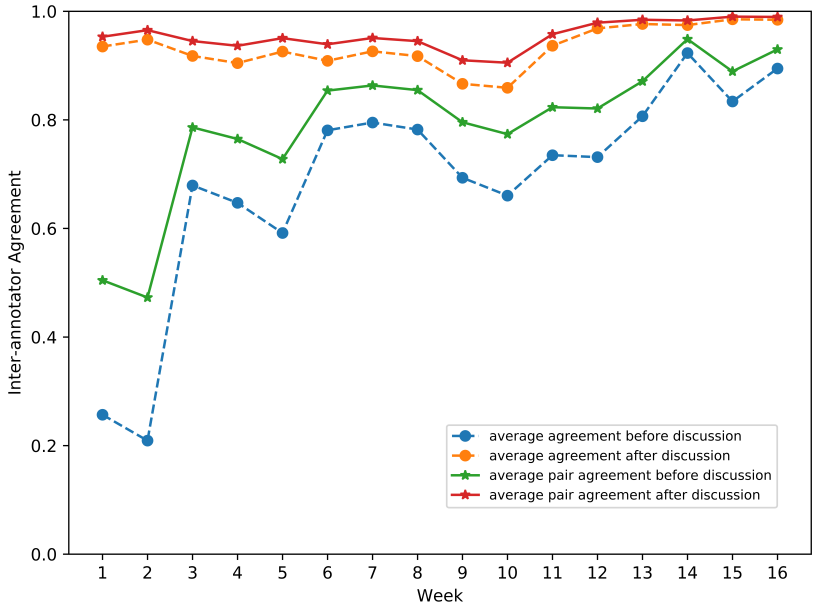

Figure 1: Inter-annotator proportion agreement results (week by week). Average agreements are the proportion agreements among three annotators. Average pair agreements are the average proportion agreements of three annotator pairs.

4 Automatic Concept Extraction

4.1 The Task Formulation

We formulated the concept extraction task in the following way: *given a textbook which has multiple chapters and each chapter includes several sections, extract a list of concepts appearing in each of the sections.*

Concept extraction task is similar to the tasks of keyphrase extraction and named entity recognition. However, it is more challenging because (1) concepts vary significantly across domains, (2) it is hard to define the boundary between the domains, and (3) there is a lack of clear signifiers and context. In order to perform this task, we recast it as a binary classification problem for a list of extracted candidates. We train a supervised learning model to classify a term or phrase candidate to be a concept or not. The details of our framework is described in the next section.

4.2 The Framework

Preprocessing: We preprocess the textbook to extract section names, titles, and the text content of each section.

Data preparation: We use Stanford's POS tagger[3] (Toutanova et al., 2003) to annotate each word in the text with its linguistic part-of-speech. Defining that a concept is a noun or noun phrase, we apply linguistic rules (e.g., '*noun* + *noun*' or '*adjective* + *noun*') using regular expression to extract all possible nouns and noun phrases in the text. We only consider unigrams, bigrams, trigrams and four-grams, which account for 99.42% of all the unique concepts (shown in Table 1).

After extracting all noun phrases, we used a stop-list to filter non-descriptive words (mostly determiners) that add no additional meaning to the concept (e.g., *such*, *same*, *many*, *little*, *few*, or *certain*). For instance, though "many searching algorithms" and "searching algorithms" both are noun phrases extracted from the text, it is very easy to recognize that "many searching algorithms" should not be considered as a concept.

Let take a look at the example below:

"*The general strategy for determining a stop list is to sort the terms by collection frequency*."

After tagging: "*The_DT general_JJ strategy_NN for_IN determining_VBG a_DT stop_NN list_NN is_VBZ to_TO sort_VB the_DT terms_NNS by_IN collection_NN frequency_NN* ._."

Final candidate list: {*general strategy*, *strategy*, *stop list*, *stop*, *list*, *terms*, *collection frequency*, *collection*, *frequency*}

Feature extraction: After obtaining the final candidate list, we extract all features for each of the candidates. The feature set includes *linguistic* features (e.g., POS, two tokens before, two tokens after), *statistics* features (e.g., term frequency,

[3]https://nlp.stanford.edu/software/tagger.shtml

tf-idf), its match to *external resources* (i.e., wikipedia titles and ACM keyword repository) and its presence in the *section title*. The details of the feature set are described in the next section.

Model training: We trained a logistic regression model on the feature vectors of candidate keyphrases. All non-binary features in our model are binned and discretized as binary features. In this way, our logistic model is capable of learning non-linear relationships with those features. For the cross evaluation purpose, we split data into 5 folds; each fold consists of 80% for training and 20% for testing. Aware of the cases that multiple candidates could be from the same phrase (e.g., '*postings list*' appearing in multiple sections of the book), we force those candidates to be only in the train set or only in the test set when splitting the data.

4.3 Features

To train our concept extractor, we use 25 types of features listed in Table 2. In total, we have 7661 features for this specific dataset. We categorize the features into four subsets – those which are *linguistic*, *statistics*-based, use *external resources* or use a *section title*. Each subset represents different identifiers and cues that could help recognize concepts.

4.3.1 Linguistic features

Linguistic features provide the most informative and significant cues to identify concepts. These features capture both *internal* (i.e., constituent words) and *external* (i.e., context) characteristics of the concept candidate.

- *POS*(features 1-5): encode the part-of-speech structure of the candidate. This set of features helps to recognize common patterns that concepts may have (e.g., *noun* + *noun*, or *adjective* + *noun*). In addition, we included separate POS features for specific tokens of the candidate, which could provide more fine-grained patterns for the extractor.

- *Context* (features 6-17): describes the surrounding context of the candidate (e.g., the first word to left and the POS of the first word to left of the candidate).

- *Length of candidate*: number of tokens the candidate has. As we can see from Table 1, the distribution of different n-grams varies significantly.

4.3.2 Statistic features

In this section, we present several statistics-based features, which are inspired by work in information retrieval. These methods (also known as term-scoring methods) give a specific value to a candidate based on how it is distributed in the textbook. The central component of term scoring is *term frequency*.

ID	Feature	Value	Description	% loss of F_1	% loss of *AUC*	F_1'	*AUC'*
1*	pos[all]	{jj_jj_nn_nn,...}	Concatenation of the POS of each of the tokens in the candidate	1.32	0.00	0.39	0.78
2	pos[0]	{nnp, nn, ...}	POS of the first token of the candidate	0.00	0.00	0.00	0.66
3	pos[1]	{nnp, nn, ...}	POS of the 2nd token of the candidate if exists	0.00	0.00	0.00	0.63
4	pos[2]	{nnp, nn, ...}	POS of the 3rd token of the candidate if exists	0.00	0.00	0.00	0.60
5	pos[3]	{nnp, nn, ...}	POS of the 4th token of the candidate if exists	0.00	0.00	0.00	0.56
6*	word[1_left]	<string>	First word *w* on the left of the candidate if exists (e.g., *the*)	3.95	1.06	0.40	0.75
7*	word[1_right]	<string>	First word *w* on the right of the candidate if exists (e.g., *is*)	2.63	1.06	0.34	0.73
8*	word[2_left]	<string>	Second word *w* on the left of the candidate if exists (e.g., *the*)	3.95	1.06	0.39	0.73
9*	word[2_right]	<string>	Second word *w* on the right of the candidate if exists (e.g., *is*)	1.32	0.00	0.31	0.66
10	word[3_left]	<string>	Third word *w* on the left of the candidate if exists (e.g., *the*)	0.00	0.00	0.00	0.51
11	word[3_right]	<string>	Third word *w* on the right of the candidate if exists (e.g., *is*)	0.00	0.00	0.00	0.51
12	pos[1_left]	{nnp, nn, ...}	POS of the first word *w* on the left of the candidate if exists	0.00	0.00	0.26	0.66
13	pos[1_right]	{nnp, nn, ...}	POS of the first word *w* on the right of the candidate if exists	0.00	0.00	0.28	0.69
14	pos[2_left]	{nnp, nn, ...}	POS of the second word *w* on the left of the candidate if exists	0.00	0.00	0.26	0.67
15	pos[2_right]	{nnp, nn, ...}	POS of the second word *w* on the right of the candidate if exists	0.00	0.00	0.26	0.65
16	pos[3_left]	{nnp, nn, ...}	POS of the third word *w* on the left of the candidate if exists	0.00	0.00	0.00	0.51
17	pos[3_right]	{nnp, nn, ...}	POS of the third word *w* on the right of the candidate if exists	0.00	0.00	0.00	0.51
18	length	{1, 2, 3, 4}	Number of tokens in the candidate	0.00	0.00	0.00	0.59
19*	fre	<numeric>	Frequency of the candidate in a section	0.00	0.00	0.36	0.65
20*	cf	<numeric>	Frequency of the candidate in the textbook	2.63	1.06	0.44	0.76
21*	tf*idf	<numeric>	tf*idf score of the candidate in the textbook	1.32	0.0	0.26	0.69
22	lang	{true, false}	Statistical testing if the candidate comes from the same distribution	0.00	0.00	0.00	0.51
23	wTitle	{true, false}	The candidate appearing in a Wikipedia title	0.00	0.00	0.00	0.59
24	acm	{true, false}	The candidate appearing in the ACM keyword repository or not	2.63	1.06	0.00	0.72
25*	sTitle	{true, false}	The candidate appearing in a section title of the textbook or not	1.32	1.06	0.47	0.67

Table 2: Features used in our concept extractor.

- *Frequency* (fre): how many times a candidate occurs in a particular section. We created binary features where the frequency is less than or equal to 1, 2, 3, 4, 5, or 6. The intuition is that if a candidate appears many times in a section, it may be a less informative but generic term.

- *Collection frequency* (cf): how many times a candidate occurs in the entire textbook. We also created a set of *cf*-related binary features where the frequency is considered up to a heuristic threshold of 50.

- *Term frequency-inversed document frequency* (tf-idf): idf is a measure of the informativeness of the candidate. A set of binary features were created for the log of tf-idf score (at various thresholds).

- *Language model* (lang): this feature is evaluated based on the probability distribution of a *foreground* corpus (i.e., Information Retrieval) and a *back-*

ground corpus (i.e., a large corpus encoding the knowledge about the world). We use the content of the textbook as the foreground corpus and calculate the distribution for each of the candidates. For the background, we obtained the distribution of n-grams from Bing Web Language Model API. We hypothesize that a candidate is more likely to be a concept if it's probability distribution in the foreground corpus is significantly higher than in the background corpus.

4.3.3 External resources

These features attempt to improve the performance of the model by exploiting existing lexical knowledge bases which are usually built by domain experts. These resources are independent from the training data. They can be computed directly without the need of labeling the training data. In this work, we leverage the following resources:

- *Wikipedia*: based on the observation that a candidate is likely to be a concept if there is an article discussing it or some of its aspects. We collected all IR-related article titles. This collection is used to check if a candidate appears in any of these Wikipedia titles. This feature is called *Wikipedia title*-based feature (or *wTitle* for short).
- *ACM Computer Science keyphrase repository*: We assume that if a candidate appears in the the collection of keywords in Computer Science domain published by ACM, it is very likely to be a concept.

4.3.4 Section titles

Book authors use section titles to inform readers of the topics, ideas or problems they are going to present. It is intuitive to assume that if a candidate appears in a section title, it should have a significant meaning contributing to the topic. Therefore, we add one more feature to the model called *section title*-based feature (or *sTitle* for short).

5 The Evaluation

5.1 The Evaluation Approach

To evaluate our model and compare it with the baselines, we use several metrics: AUC, micro *precision*, micro *recall*, micro F_1, macro *precision*, macro *recall* and macro F_1. We compute the scores using *exact matching*. While we are aware of the limitation of *exact matching* for keyphrase extraction evaluation, it is still the best solution for comparing models' performance without humans in the loop.

5.2 Baselines

We compare our model with the following baselines.

1. **Random model**: The random model mimics the process of building a logistic regressor without training any model; it randomly assign probabilities from 0 to 1 to candidates and use the cutoff of 0.35 (i.e., the same as the main model) to classify concepts.

2. **Linguistics model**: The logistic regression model only uses the linguistic-based features (i.e., features 1-18), also used in (Yih et al., 2006; Nguyen and Kan, 2007)

3. **Statistics model**: The logistic regression model, which uses only the statistics-based features (i.e., features 19-22), also used in (Hammouda et al., 2005; Witten et al., 1999; Rose et al., 2010; Nguyen and Kan, 2007; Yih et al., 2006).

4. **External resource baseline**: The logistic regression model, which uses only the External resource-based features (i.e., features 23-24), also used in (Wang et al., 2015).

5. **Title baseline**: The logistic regression model, which uses only the title-based features (i.e., 24), also used in (Wang et al., 2015; Yih et al., 2006) for extracting concepts and (Labutov et al., 2017) for predicting prerequisite and outcome concepts.

6. **TextRank baseline**: a well-known graph-based approache for keyphrase extraction (Mihalcea and Tarau, 2004).

7. **TopicRank baseline**: a graph-based ranking method to discover topical representations for documents from which keyphrases are generated (Bougouin et al., 2013).

8. **Rapid automatic keyword extraction (RAKE) baseline**: an unsupervised, domain independent and language independent approach for extracting keywords from individual documents (Rose et al., 2010).

9. **IBM Natural Language Understanding API baseline**: the client library *watson_developer_cloud* provided for Python[4]. Given a text document, the API will return a list of keywords or entities.

10. **CopyRNN baseline**: a RNN-based model using encoder-decoder architecture to predict keyphrases (Meng et al., 2017).

[4]https://github.com/watson-developer-cloud/python-sdk

System	AUC	Micro p	Micro r	Micro F_1	Macro p	Macro r	Macro F_1
Baseline 1: Random (*)	0.50	0.20	0.67	0.31	0.14	0.50	0.21
Baseline 2: Linguistics (*)	0.90	0.66	0.63	0.65	0.47	0.57	0.51
Baseline 3: Statistics (*)	0.80	0.55	0.58	0.56	0.50	0.25	0.34
Baseline 4: External resources (*)	0.72	0.36	0.47	0.41	0.29	0.44	0.35
Baseline 5: Titles (*)	0.67	0.55	0.42	0.47	0.43	0.14	0.21
Baseline 6: TextRank	–	0.17	0.10	0.17	0.11	0.10	0.10
Baseline 7: TopicRank	–	0.16	0.16	0.16	0.11	0.28	0.16
Baseline 8: RAKE	–	0.15	0.13	0.14	0.07	**0.63**	0.12
Baseline 9: IMB API	–	0.25	0.19	0.22	0.16	0.26	0.20
Baseline 10: CopyRNN	–	0.23	0.22	0.23	0.26	0.20	0.23
Baseline 11: Humans (AMT)	–	0.40	0.38	0.39	0.29	0.55	0.38
Our system	**0.94**	**0.75**	**0.77**	**0.76**	**0.61**	0.58	**0.60**

* p<<0.01 (paired nonparametric McNemar test), p: precision, r: recall

Table 3: AUC, micro F_1 and macro F_1 of our model compared to the baselines. Significance testing only performed on the random and partial models.

11. **Humans/AMT baseline**: We recruited three annotators from Amazon Mechanical Turk[5]. The annotators were assigned to chapter 6 and 8 of IIR book, including 13 sections (i.e., we chose these two chapters based on the reasonable amount of text for the annotation assignments).

5.3 Results

With this set of features, the logistic regression model with 5-fold cross validation achieved an AUC score of 0.94 and a micro F_1 score of 0.76 (see Table 3) for the concept classification task. It is significantly better than all of the partial models (i.e., with different subsets of features). Among the partial models, the *linguistic* model performs best. It means that for the task of concept classification from textbooks, language-based features taking advantage of the syntactic structure and the context of candidates provide the most important signifiers. The *statistics* model also achieves a significant result.

Table 3 shows that our model outperforms all the baselines. Although RAKE achieves the highest macro recall of 63%, its precision is the lowest (and a lower F_1 score as a result). Again, *linguistic* model is the best among the partial models, achieving the F_1 of 0.51. It is also significantly better than the other baselines including human (i.e., Mechanical Turk).

CopyRNN, deep neural net-based model, do not perform well for this task as expected. This is likely due to the fact that we directly use the original model which was trained on a different dataset (the paper abstract-keyword datasets) to predict concepts in the textbook.

Our model can achieve a precision as high as 98% (at 19% recall) or recall as high as 97% (at 37% precision), depending on the preference of the user (see Figure 2). Since concept extraction still remains a very challenging task, it is difficult to accomplish a high recall and a high precision at the same time. The availability

[5]https://www.mturk.com

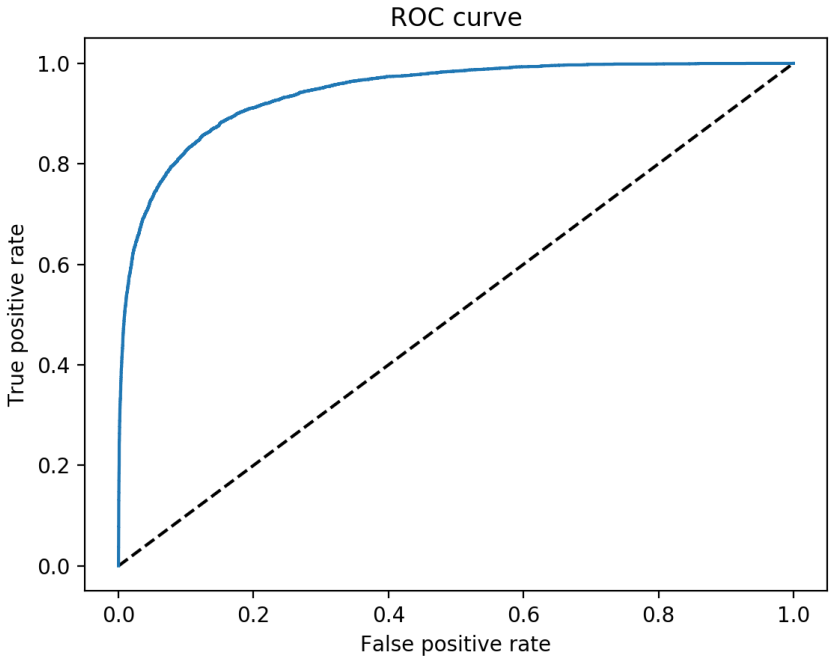

Figure 2: ROC curve from the main concept classifier.

to choose between a high recall or high precision could help to improve downstream tasks depending on what is more important. In document linking task, for instance, we may want to achieve a high recall (i.e., identify many concepts) to distinguish documents. On the other hand, a high precision could result in better performance for student modeling and prediction tasks, which requires more precise and accurate concepts.

5.4 Error Analysis

Errors propogate to the final prediction stage from multiple sources. Some come from the preprocessing step due to noisy text; others are from the model itself. After a very careful data preprocessing and preparation, we were be able to obtain 97.72% of all the expert-annotated concepts; most of missing concepts come from special characters (e.g., *(pseudo-)relevance feedback)* or errors of POS tagging.

As we can see from Table 4, the model failed to identify most of the 4-gram concepts. 57% of unigrams were not recognized, accounting for more than half of false negative cases. On the other hand, there are predicted concepts from the model that could be considered as concepts but were not annotated by the experts; for example, *optimization*, *bayesian network*, *frequency-based feature selection*, *multinomial unigram language model*.

Some of errors come from partial matching; for instance, *maximum likelihood estimates* is an actual concept, and the model predicts *maximum likelihood esti-*

	Number of concepts	% of ground truth concepts
1-grams	639	57%
2-grams	367	23%
3-grams	162	38%
4-grams	43	89%
Total	1211	38%

Table 4: Concepts annotated by experts but not predicted by the model (false negative).

mates as a concept.

For the candidates predicted by the model but not annotated by the experts (i.e., false positive), we had an expert to additionally evaluate them. There are 13%, 30% and 30% of unigrams, bigrams and trigrams respectively which could be considered as concepts based on the expert's judgement. Those cases come from either the experts missing them during the annotation process or partial matching.

For both the false negative or false positive cases, we can see that unigram candidates and concepts contribute to most of the failed cases, meaning that it's harder to deal with unigram concepts compared to bigrams or trigrams. Moreover, as can be seen in Table 4, there are only 23% of actual bigram concepts that are not identified by the model. Thought bigram concepts account for 56.45% of all the concepts, they are much easier to recognize.

6 Conclusions and Future Work

In this paper, we present a thorough, rigorous and systematic analysis of a supervised learning approach for the task of concept extraction from educational content. We evaluate the proposed model with a newly constructed dataset by comparing with an extensive number of keyphrase extraction models.

This work is a step towards the ultimate goal of developing a new generation of intelligent textbooks. There is still room to improve the model, for example by focusing on tackling uni-gram concepts which currently have the highest error rate. Another direction for work is utilizing deep neural networks to enhance the highly engineered feature sets presented in this work. Our priority is to investigate how the outcomes of the current model could help improve downstream educational tasks such as content recommendation, and to investigate the sensitivities of these downstream tasks to the different levels of precision and recall of concept keyword extraction. We believe that the work presented in this paper will help the research community towards building the next generation learning platforms for the web.

References

Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. 2017. https://doi.org/10.18653/v1/S17-2091 Semeval 2017 task 10: Scienceie - extracting keyphrases and relations from scientific publications. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 546–555. Association for Computational Linguistics.

Adrien Bougouin, Florian Boudin, and Béatrice Daille. 2013. http://aclweb.org/anthology/I13-1062 Topicrank: Graph-based topic ranking for keyphrase extraction. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 543–551, Nagoya, Japan. Asian Federation of Natural Language Processing.

Corina Florescu and Cornelia Caragea. 2017. https://doi.org/10.18653/v1/P17-1102 Positionrank: An unsupervised approach to keyphrase extraction from scholarly documents. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1115. Association for Computational Linguistics.

Maria Grineva, Maxim Grinev, and Dmitry Lizorkin. 2009. https://doi.org/10.1145/1526709.1526798 Extracting key terms from noisy and multitheme documents. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 661–670, New York, NY, USA. ACM.

Khaled M. Hammouda, Diego N. Matute, and Mohamed S. Kamel. 2005. Corephrase: Keyphrase extraction for document clustering. In *Machine Learning and Data Mining in Pattern Recognition*, pages 265–274, Berlin, Heidelberg. Springer Berlin Heidelberg.

Anette Hulth. 2003. https://doi.org/10.3115/1119355.1119383 Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, EMNLP '03, pages 216–223, Stroudsburg, PA, USA. Association for Computational Linguistics.

Xin Jiang, Yunhua Hu, and Hang Li. 2009. https://doi.org/10.1145/1571941.1572113 A ranking approach to keyphrase extraction. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 756–757, New York, NY, USA. ACM.

Igor Labutov, Yun Huang, Peter Brusilovsky, and Daqing He. 2017. https://doi.org/10.1145/3097983.3098187 Semi-supervised techniques for mining learning outcomes and prerequisites. In *Proceedings of the 23rd ACM*

SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '17, pages 907–915, New York, NY, USA. ACM.

Igor Labutov and Hod Lipson. 2016. Web as a textbook: Curating targeted learning paths through the heterogeneous learning resources on the web. In *Proceedings of the 9th International Conference on Educational Data Mining*, EDM '16, pages 110–118.

Tho Thi Ngoc Le, Minh Le Nguyen, and Akira Shimazu. 2016. Unsupervised keyphrase extraction: Introducing new kinds of words to keyphrases. In *AI 2016: Advances in Artificial Intelligence*, pages 665–671, Cham. Springer International Publishing.

Feifan Liu, Deana Pennell, Fei Liu, and Yang Liu. 2009a. http://dl.acm.org/citation.cfm?id=1620754.1620845 Unsupervised approaches for automatic keyword extraction using meeting transcripts. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 620–628, Stroudsburg, PA, USA. Association for Computational Linguistics.

Zhiyuan Liu, Wenyi Huang, Yabin Zheng, and Maosong Sun. 2010. http://dl.acm.org/citation.cfm?id=1870658.1870694 Automatic keyphrase extraction via topic decomposition. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 366–376, Stroudsburg, PA, USA. Association for Computational Linguistics.

Zhiyuan Liu, Peng Li, Yabin Zheng, and Maosong Sun. 2009b. http://dl.acm.org/citation.cfm?id=1699510.1699544 Clustering to find exemplar terms for keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 257–266, Stroudsburg, PA, USA. Association for Computational Linguistics.

Olena Medelyan, Eibe Frank, and Ian H. Witten. 2009. http://dl.acm.org/citation.cfm?id=1699648.1699678 Human-competitive tagging using automatic keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3 - Volume 3*, EMNLP '09, pages 1318–1327, Stroudsburg, PA, USA. Association for Computational Linguistics.

Rui Meng, Sanqiang Zhao, Shuguang Han, Daqing He, Peter Brusilovsky, and Yu Chi. 2017. http://arxiv.org/abs/1704.06879 Deep keyphrase generation. *CoRR*, abs/1704.06879.

Rada. Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into texts. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, Barcelonaand Spain.

Thuy Dung Nguyen and Min-Yen Kan. 2007. Keyphrase extraction in scientific publications. In *Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers*, pages 317–326, Berlin, Heidelberg. Springer Berlin Heidelberg.

Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. Automatic keyword extraction from individual documents. In *Text Mining: Applications and Theory*, pages 1–20. John Wiley and Sons, Ltd.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. https://doi.org/10.3115/1073445.1073478 Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 173–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Xiaojun Wan and Jianguo Xiao. 2008. http://aclweb.org/anthology/C08-1122 Collabrank: Towards a collaborative approach to single-document keyphrase extraction. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 969–976. Coling 2008 Organizing Committee.

Shuting Wang, Chen Liang, Zhaohui Wu, Kyle Williams, Bart Pursel, Benjamin Brautigam, Sherwyn Saul, Hannah Williams, Kyle Bowen, and C. Lee Giles. 2015. https://doi.org/10.1145/2682571.2797062 Concept hierarchy extraction from textbooks. In *Proceedings of the 2015 ACM Symposium on Document Engineering*, DocEng '15, pages 147–156, New York, NY, USA. ACM.

Ian H. Witten, Gordon W. Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevill-Manning. 1999. https://doi.org/10.1145/313238.313437 Kea: Practical automatic keyphrase extraction. In *Proceedings of the Fourth ACM Conference on Digital Libraries*, DL '99, pages 254–255, New York, NY, USA. ACM.

Yiming Yang, Hanxiao Liu, Jaime Carbonell, and Wanli Ma. 2015. https://doi.org/10.1145/2684822.2685292 Concept graph learning from educational data. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM '15, pages 159–168, New York, NY, USA. ACM.

Wen-tau Yih, Joshua Goodman, and Vitor R. Carvalho. 2006. https://doi.org/10.1145/1135777.1135813 Finding advertising keywords on web pages. In *Proceedings of the 15th International Conference on World Wide Web*, WWW '06, pages 213–222, New York, NY, USA. ACM.