Keyphrase Extraction in Scholarly Digital Library Search Engines

Krutarth Patel[1], Cornelia Caragea[2], Jian Wu[3], and C. Lee Giles[4]

[1] Computer Science, Kansas State University
kipatel@ksu.edu
[2] Computer Science, University of Illinois at Chicago
cornelia@uic.edu
[3] Computer Science, Old Dominion University
jwu@cs.odu.edu
[4] Information Sciences and Technology, Pennsylvania State University
giles@ist.psu.edu

Abstract. Scholarly digital libraries provide access to scientific publications and comprise useful resources for researchers who search for literature on specific subject areas. CiteSeerX is an example of such a digital library search engine that provides access to more than 10 million academic documents and has nearly one million users and three million hits per day. Artificial Intelligence (AI) technologies are used in many components of CiteSeerX including Web crawling, document ingestion, and metadata extraction. CiteSeerX also uses an unsupervised algorithm called noun phrase chunking (NP-Chunking) to extract keyphrases out of documents. However, often NP-Chunking extracts many unimportant noun phrases. In this paper, we investigate and contrast three supervised keyphrase extraction models to explore their deployment in CiteSeerX for extracting high quality keyphrases. To perform user evaluations on the keyphrases predicted by different models, we integrate a voting interface into CiteSeerX. We show the development and deployment of the keyphrase extraction models and the maintenance requirements.

Keywords: Scholarly Digital libraries · Keyphrase extraction · Information extraction.

1 Introduction

Online scholarly digital libraries usually contain millions of scientific documents [30]. For example, Google Scholar is estimated to have more than 160 million documents [39] Open access digital libraries have witnessed a rapid growth in their document collections as well in the past years [31]. For example, CiteSeerX's collection increased from 1.4 million to more than 10 million within the last decade. On one hand, these rapidly-growing scholarly document collections offer rich domain specific information for knowledge discovery, but, on the other hand, they pose many challenges to navigate and search for useful information in these collections.

Keyphrases of scientific papers provide important topical information about the papers in a highly concise form and are crucial for understanding the evolution of ideas in a scientific field [22, 47, 29]. In addition, keyphrases play a unique role in many downstream applications such as finding good index terms for papers [43], summarizing scientific papers [42, 41, 2], suggesting keywords in query formulation and expansion [46], recommending papers to readers [27], identifying reviewers for paper submissions [5], and clustering papers for fast retrieval [23]. Due to the high importance of keyphrases, several online digital libraries such as the ACM Digital Library have started to impose the requirement for author-supplied keyphrases. Specifically, these libraries require authors to provide keyphrases that best describe their papers. However, keyphrases have not been integrated into all sharing mechanisms. For example, the AAAI digital library (http://www.aaai.org/) does not provide keyphrases associated with the papers published in the AAAI conferences. In an effort to understand the coverage of papers with author-supplied keyphrases in open access scholarly digital libraries, we performed the following analysis: we randomly sampled 2,000 papers from CiteSeerX, and manually inspected each paper to determine whether a paper contains author-supplied keyphrases and if the paper was published by ACM. Note that in most of the ACM conference proceeding templates, the authors need to provide keyphrases (keywords) after the "Abstract" section. For completeness, the ACM templates from years 1998, 2010, 2015, and 2017 were adopted for visual inspection. Out of our 2,000 sample, only 31 (1.5%) papers were written using ACM templates and only 769 papers (38%) contain author-supplied keyphrases. Out of 31 papers written using ACM templates, 25 contain author-supplied keyphrases. The fact that around 62% of papers sampled do not have author-supplied keyphrases indicates that automatic keyphrase extraction is needed for scholarly digital libraries.

To date, many methods on the keyphrase extraction task have been proposed that perform better than NP-chunking or *tf-idf* ranking. Such methods include KEA [16], Hulth [28], TextRank [37], Maui [36], CiteTextRank [19], ExpandRank [50], CeKE [9], PositionRank [15], Key2Vec [35], BiLSTM-CRF [4], and CRFs based on word embeddings and document specific features [40]. However, keyphrase extraction has not been integrated into open access digital libraries. Most existing scholarly digital libraries [54] such as Google Scholar and Microsoft Academic do not display keyphrases. Recently, SemanticScholar started to display keyphrase-like terms called "topics." The CiteSeerX website currently displays keyphrases extracted using an unsupervised phrase chunking method [12].

In this application paper, we first review keyphrase extraction in scholarly digital libraries, using CiteSeerX as a case study. We investigate the impact of displaying keyphrases on promoting paper downloading by analyzing search engine access logs in three years from 2016 to 2018. Then, we interrogate the quality of several supervised keyphrase extraction models to explore their deployment in CiteSeerX and perform a large scale keyphrase extraction - first of its kind for this task. Moreover, to get user evaluations on the predicted keyphrases on a

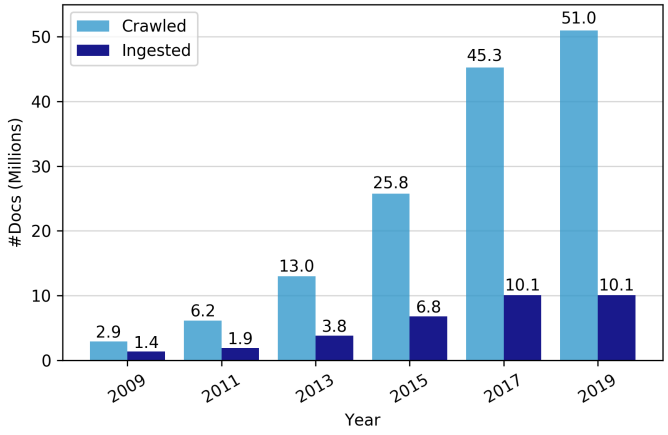


Fig. 1. Number of documents crawled and ingested from past few years in CiteSeerX.

large scale, we implement and integrate a voting interface, which is widely used in social networks and multimedia websites, such as Facebook and YouTube. We show the development and deployment requirements of the keyphrase extraction models and the maintenance requirements.

2 CiteSeerX Overview and Motivation

There are in general two types of digital library search engines. The first type obtains publications and metadata from publishers, such as ACM Digital Library, IEEE Xplore, and Elsevier. The other type, such as CiteSeerX [17], crawls the public Web for scholarly documents and *automatically* extracts metadata from these documents.

CiteSeer was launched in 1998 [18] and its successor CiteSeerX [55] has been online since 2008. Since then, the document collection has been steadily growing (see Figure 1). The goal of CiteSeerX is to improve the dissemination of and access to academic and scientific literature. Currently, CiteSeerX has 3 million unique users world-wide and is hit 3 million times a day. CiteSeerX reaches about 180 million downloads annually [48]. Besides search capabilities, CiteSeerX also provides an Open Archives Initiative (OAI) protocol for metadata harvesting. CiteSeerX receives about 5,000 requests per month to access the OAI service. Researchers are interested in more than just CiteSeerX metadata. For example, CiteSeerX receives about 10 requests for data per month via the contact form on the front page [51]. These requests include graduate students seeking project datasets and researchers that were looking for large datasets for experiments. CiteSeerX hosts a dump of the database and other data on Google Drive.

In the early stage, the crawl seeds were mostly homepages of scholars in computer and information sciences and engineering (CISE). In the past decade,

Year	#Docs. (Millions)	#Keyphrase-Clicks (Millions)	#Unique-Keyphrases (Millions)
2016	8.44	4.41	1.60
2017	10.1	7.17	1.86
2018	10.1	7.52	1.74

Table 1. The number of full text documents, the total number of keyphrase-clicks, and unique keyphrases clicked for years 2016, 2017, and 2018 in CiteSeerX.

CiteSeerX added to the crawls seed URLs from the Microsoft Academic Graph [45], and directly incorporated PDFs from PubMed, arXiv, and digital repositories in a diverse spectrum of disciplines. A recent work on subject category classification of scientific papers estimated that the fractions of papers in physics, chemistry, biology, materials science, and computer science are 11.4%, 12.4%, 18.6%, 5.4%, and 7.6%, respectively [52]. CiteSeerX is increasing its document collection by actively crawling the Web using new policies and seeds to incorporate new domains. We expect this to encourage users from multiple disciplines to search and download academic papers and to be useful for studying cross discipline citation and social networks.

Since CiteSeerX was developed, many artificial intelligence techniques have been developed and deployed in CiteSeerX [55], including but not limited to header extraction [24], citation extraction [13], document type classification [11], author name disambiguation [49], and data cleansing [44]. In addition, an unsupervised NP-Chunking method is deployed for automatic keyphrase extraction. Besides author-submitted keyphrases, CiteSeerX extracts on average 16 keyphrases per paper using NP-Chunking. Users can search for a particular keyphrase by clicking it. This feature provides a shortcut for users to explore scholarly papers in related topics of the current paper they are browsing. All automatically extracted keyphrases are displayed on the summary page, and they deliver detailed domain knowledge in scholarly documents. Every time a keyphrase is clicked, CiteSeerX searches the clicked keyphrase and refreshes the search results. To understand how keyphrases promote paper browsing and downloading, we analyze the access logs retrieved from three web servers from 2016 to 2018.

2.1 Click-log Analysis

Table 1 shows the total number of documents, keyphrase clicks, and unique keyphrases clicked from 2016 to 2018. The total number of keyphrase clicks increased significantly by $\sim$ 63% from 2016 to 2017. For years 2017 and 2018, although the total number of documents stayed about the same (10.1 million), the total number of keyphrase clicks increased by 5%. Although there is a slight decrease in the number of unique keyphrases clicked, the increase in the number of keyphrase clicks from year 2016 to year 2018 showcases the increasing use and the popularity of keyphrases.

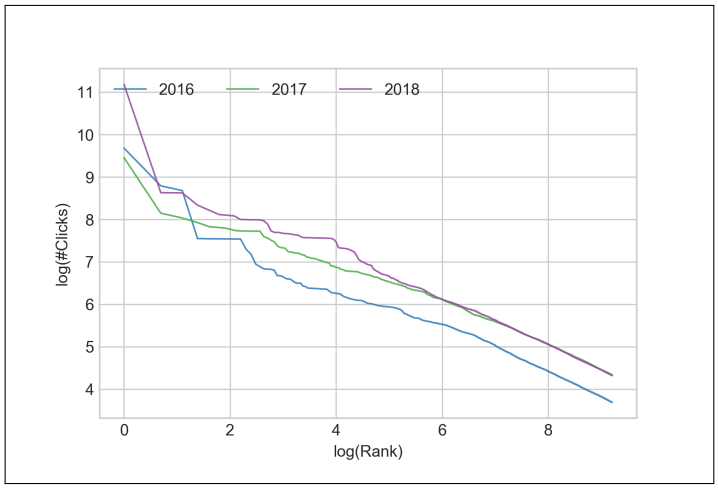


Fig. 2. log(Rank) vs log(Clicks) for top-10, 000 keyphrases clicked by users of CiteSeerX during years 2016, 2017, and 2018.

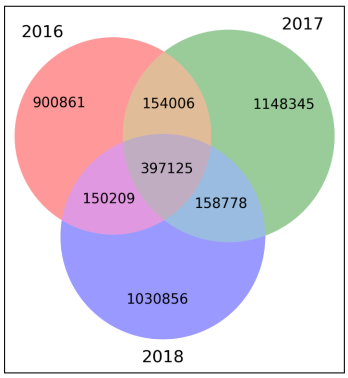


Fig. 3. Venn Diagram for all 3 years based on unique keyphrases.

Figure 2 shows the ranking versus the number of clicks (#clicks) in logarithmic scale for the 10,000 most popular keyphrases during the three years. We can see that the #click decreases exponentially as the rank increases, which mimics the Zipf's law for all three years.

Figure 3 shows the Venn diagram for the unique keyphrases clicked during years 2016, 2017, and 2018. As seen from the figure, in two consecutive years, only about one third of the keyphrases are common, whereas two third of the keyphrases are new. For example, 1.6 million unique keyphrases were clicked in 2016 but only about $551k$ (33%) were carried to 2017. Similarly, 1.86 million unique keyphrases were clicked in 2017, but only $555k$ (30%) were carried out in 2018. This trend implies that user interests have been rapidly evolving over these years, but there is still a considerable number of topics searched among

Year	Keywords
2016	DgNe, local, bullying, violence, bullied, bully, aggressive, aggression, R. Nobrega, experimental result, data, wide range, machine, lpEu, dvd, last year, recent year, artificial intelligence, key word, new technology
2017	key word, experimental result, wide range, large number, string theory, bullying, violence, bullied, bully, aggressive, aggression, recent year, new method, artificial intelligence, important role, machine learning, neural network, online version, environmental protection agency, wide variety
2018	JMQi, experimental result, key word, large number, wide range, aggression, violence, bullying, bully, bullied, aggressive, recent year, case study, wide variety, different type, sustainable development, informational security, VWBc, sensor network, simulation result

Table 2. Top-20 keyphrases clicked during years 2016, 2017, and 2018.

several years. These conclusions are made based on the analysis of open-access documents from a three years time period. However, further analysis is needed for more comprehensive conclusions.

Table 2 shows the top-20 most frequent keyphrases clicked. We can see that the extracted keyphrases are not always terminological concepts as seen usually in author-submitted keyphrases. Examples such as "local", "experimental results", "wide range", and "recent year" were extracted just because they are noun phrases. This indicates that more sophisticated models are necessary to improve the quality of extracted keyphrases. It is interesting that these phrases were highly clicked, but investigating the reason is beyond the scope of this paper.

3 AI-Enabled Keyphrase Extraction

Here we describe three supervised keyphrase extraction models that we explore to integrate into CiteSeerX: KEA [16], Hulth [28], and Citation-enhanced Keyphrase Extraction (CeKE) [9]. Unlike KEA and Hulth, which only use the title and abstract of a given research article, CeKE exploits citation contexts along with the title and abstract of the given document. A citation context is defined as the text within a window of n words surrounding a citation mention. A citation context includes cited and citing contexts. A citing context for a target paper p is a context in which p is citing another paper. A cited context for a target paper p is a context in which p is cited by another paper. For a target paper, all cited contexts and citing contexts are aggregated into a single context. Figure 4 shows an example of a small citation network using a paper (Paper 1) and its citation network neighbors. We can see the large overlap between the authors-submitted keyphrases and the citation contexts.

KEA: Frank et al. [16] used statistical features for the keyphrase extraction task and proposed a method named KEA. KEA uses following statistical features: *tf-idf*, i.e., the term frequency - inverse document frequency of a candidate

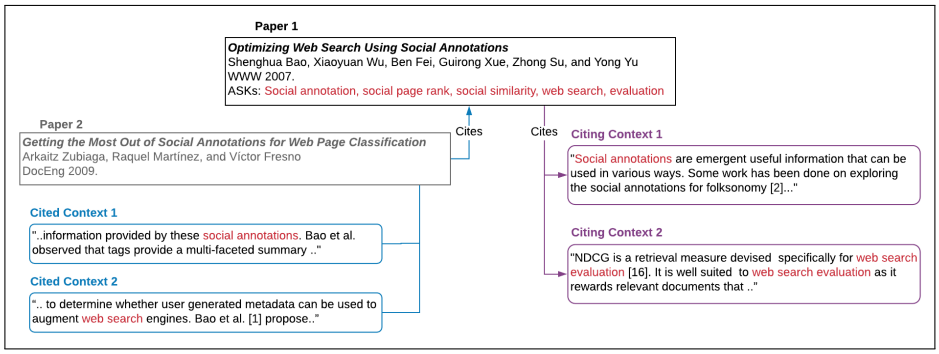


Fig. 4. A small citation network for Paper 1.

phrase and the *relative position* of a candidate phrase, i.e., the position of the first occurrence of a phrase normalized by the number of words of the target paper. KEA extracts keyphrases from the title and abstract of a given paper.

Hulth: Hulth [28] argued that adding linguistic knowledge such as syntactic features can yield better results than relying only on statistics such as a term frequency (*tf*) and *n*-grams. Hulth showed remarkable improvement by adding part-of-speech (POS) tag as a feature along with statistical features. The features used in Hulth's approach are *tf*, *cf* (i.e., collection frequency), *relative position* and *POS tags* (if a phrase is composed by more than one word, then the POS will contain the tags of all words). Similar to KEA, Hulth extracts keyphrases only from the title and abstracts.

Citation-Enhanced Keyphrase Extraction (CeKE): Caragea et al. [9] proposed CeKE and showed that the information from the citation network in conjunction with traditional frequency-based and syntactical features improves the performance of the keyphrase extraction models.

CeKE uses the following features: *tf-idf*; *relative position*; POS tags of all the words in a phrase; *first position* of a candidate phrase, i.e., the distance of the first occurrence of a phrase from the beginning of a paper; *tf-idf-Over*, i.e., a boolean feature, which is true if the *tf-idf* of a candidate phrase is greater than a threshold θ; *firstPosUnder*, also a boolean feature, which is true if the distance of the first occurrence of a phrase from the beginning of a target paper is below a certain threshold β. *Citation Network based features* include: *inCited* and *inCiting*, i.e., boolean features that are true if the candidate phrase occurs in cited and citing contexts, respectively; and *citation tf-idf*, i.e., the *tf-idf* score of each phrase computed from the aggregated citation contexts.

In our experiments, we compare three variants of CeKE: CeKE-Target that uses only the text from the target document; CeKE-Citing that uses the text from the target document and its citing contexts; CeKE-Cited that uses the text from the target document and its cited contexts; and CeKE-Both that uses both types of contexts.

ACM-CiteSeerX-KE					
Num. (#) Papers	Avg. # keyphrases	# keyphrases			
		#unigrams	#bigrams	#trigrams	# > trigrams
1,846	3.79	3,027	3,015	871	83

Table 3. The dataset description.

4 Experiments and Results

In this section, we first describe the dataset used for training and testing the keyphrase extraction models, the process of finding candidate phrases, and then present experimental results.

4.1 Dataset

We matched 30,000 randomly selected ACM papers against all CiteSeerX papers by title and found 6,942 matches. Among these papers, 6,942, 5,743, and 5,743 papers have citing, cited, and both types of contexts, respectively. To create a dataset, we consider the documents for which we have both types of contexts and at least 3 author-supplied keyphrases appearing in titles or abstracts. We name this dataset as **ACM-CiteSeerX-KE**. Using these criteria, we identified 1,846 papers, which we used as our dataset for evaluation. The gold-standard contains the author-supplied keyphrases present in a paper (its title and abstract). Table 3 shows a summary of **ACM-CiteSeerX-KE** and contains the number of papers in the dataset, the average number of author-supplied keyphrases, and the number of n-gram author-supplied keyphrases, for $n = 1, 2, 3$, and $n > 3$.

4.2 Generating Candidate Phrases

We generate candidate phrases for each document by applying POS filters. Consistent with previous works [9, 28, 32, 37, 50], these candidate phrases are identified using POS-tags of words, consisting of only nouns and adjectives. We apply Porter stemmer on each word. The initial position of each word is kept before removing any words. Second, to generate candidate phrases, contiguous words extracted in the first step are merged into n-grams ($n = 1, 2, 3$). Finally, we eliminate candidate phrases that end with an adjective and unigrams that are adjectives [9, 50].

Evaluation metrics. To evaluate the performance of the keyphrase extraction methods, we use the following metrics: precision, recall and F1-score for the positive class since the correct identification of positive examples (keyphrases) is more important. These metrics are widely used in previous works [9, 28, 37, 50]. The reported values are averaged in 10-fold cross-validation experiments, where folds were created at document level and candidate phrases were extracted from the documents in each fold to form the training and test sets. In all experiments, we used Naïve Bayes on the feature vectors extracted by each model.

Model	Pr (%)	Re (%)	F1 (%)	Time/Doc (Sec)
NP-Chunking	04.26	29.19	07.44	**1.01**
Hulth	25.91	16.15	19.86	4.47
KEA	**30.41**	20.78	24.65	4.53
CeKE-Target	27.31	35.57	30.86	4.69
CeKE-Citing	25.65	40.45	31.37	6.61
CeKE-Cited	26.49	**42.73**	**32.68**	7.14
CeKE-Both	25.07	42.19	31.42	7.97

Table 4. The comparison of different models using 10-fold cross-validation on **ACM-CiteSeerX-KE**.

4.3 Results and Discussion

Table 4 shows the performance of NP-Chunking, KEA, Hulth, CeKE-Target, CeKE-Citing, CeKE-Cited, and CeKE-Both. The table shows the evaluation measures and time taken by each method using 10-fold cross-validation on **ACM-CiteSeerX-KE**. In NP-Chunking, the given text is first tokenized and tagged by a POS tagger. Based on the POS-tagging result, a grammar-based chunk parser is applied to separate two types of phrase chunks: (1) nouns or adjectives, followed by nouns (e.g., "relational database" or "support vector machine"), and (2) two chunks of (1) connected with a preposition or conjunction (e.g., "strong law of large numbers"). Time is measured on a computer with Xenon E5-2630 v4 processor and 32GB RAM. In CiteSeerX, the header extraction tool can extract the title, abstract, and citing contexts for a target document. However, to extract cited contexts in CiteSeerX, there is an overhead of 1.2 seconds per document on average to search and extract it from the CiteSeerX database.

It can be seen from Table 4 that, CeKE-Cited achieves the highest recall and F1 of 42.73% and 32.68%, respectively. KEA achieves the highest precision of 30.41% compared with other models with top-5 predictions. NP-Chunking takes the shortest time of 1.01 seconds to extract keyphrases from a document. However, NP-Chunking suffers from low precision and F1. CeKE variants outperform Hulth and KEA in terms of recall and F1, i.e., CeKE-Citing achieves an F1 of 32.68% as compared with 24.65% achieved by KEA. Moreover, CeKE variants that make use of citation contexts outperform CeKE-Target that does not use any citation contexts.

It can be seen from the table that CeKE-Cited achieves highest F1 of 32.68%. However, CeKE-Citing takes less time compared with CeKE-Cited, i.e., CeKE-Citing takes 6.61 seconds on average per document compared with 7.14 seconds taken by CeKE-Cited. CeKE-Citing and CeKE-Both achieve comparable F1 of 31.37% and 31.42%, respectively. In terms of speed, CeKE-Target is the fastest among other variants because it does not need to perform POS tagging for citation contexts. Citing contexts can be extracted relatively straightforward from the content of the document. On the other hand, to extract cited contexts, we need the citation graph, from which we can obtain documents citing the target

Title: Incorporating **site-level knowledge** to extract structured data from **web forums**

Abstract: **Web forums** have become an important data resource for many web applications, but extracting **structured data** from unstructured **web forum** pages is still a challenging task [...]. In this paper, we study the problem of **structured data** extraction from various **web forum** sites. Our target is to find a solution as general as possible to extract **structured data**, such as post title, post author, post time, and post content from any forum site. In contrast to most existing **information extraction** methods, which only leverage the knowledge inside an individual page, we incorporate both page-level and **site-level knowledge** and employ **Markov logic networks** (MLNs) [...]. The experimental results on 20 forums show a very encouraging **information extraction** performance, and demonstrate the ability of the proposed approach on various forums. [...]

Author-supplied keyphrases: *Web forums, Structured data, Information extraction, Site level knowledge, Markov logic networks*

CeKE-Citing predicted keyphrases: ***web forum***, ***Site Level Knowledge***, *forum,* ***structured data***
Hulth predicted keyphrases: *forum, page, Knowledge, post,* ***Site Level Knowledge***, ***web forum***, ***structured data***
KEA predicted keyphrases: ***Site Level Knowledge***, ***web forum***, *forum, post*

Fig. 5. The title, abstract, author-supplied keyphrases and predicted keyphrases of an ACM paper. The phrases marked with cyan in the title and abstract shown in the figure are author-supplied keyphrases.

paper. We plan to select CeKE-Citing to deploy along with Hulth and KEA for the following reasons: CeKE-citing is faster than CeKE-cited and CeKE-Both; extracting cited contexts has an extra overhead to find it within a citation network; and cited context may not be present for all the articles.

Anecdotal Example: To demonstrate the quality of extracted phrases by different methods (CeKE-Citing, Hulth, and KEA), we select an ACM paper at random from the testing corpus and manually compared the keyphrases extracted by the three methods and the author-supplied keyphrases (Figure 5). Specifically, the cyan bold phrases shown in the text on the top of the figure represent author-supplied keyphrases, whereas the bottom of the figure shows author-supplied keyphrases and predicted keyphrases by each evaluated model. It can be seen from the figure that the CeKE-Citing predicted four keyphrases out of which three are ASKs. Hulth predicted seven keyphrases out of which three are author-supplied keyphrases. KEA predicted three keyphrases out of which two belong to author-supplied keyphrases. The predicted keyphrases by all three models that do not belong to author-supplied keyphrases are single words. This example demonstrates that CeKE-citing exhibits a better performance than the other two models.

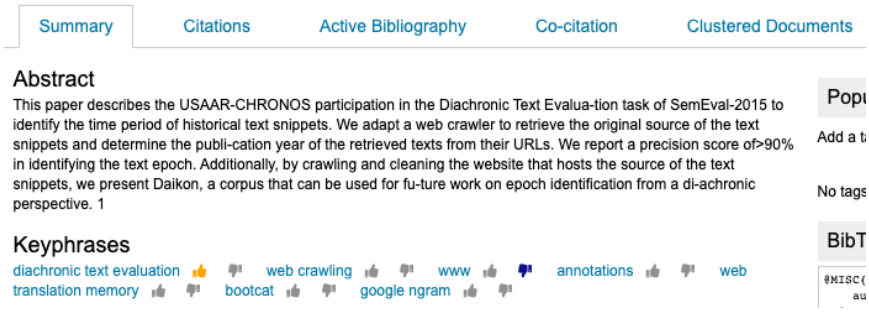


Fig. 6. A clip of a portion of a CiteSeerX paper's summary page containing a "Keyphrase" section that displays keyphrases extracted. Each keyphrase has a thumbup and a thumbdown button. A logged in user can vote by clicking these buttons.

5 Crowd-sourcing

The comparison between different keyphrase extraction models relies on ground truth datasets compiled from a small number of papers. We propose to evaluate keyphrase extraction models using crowd-sourcing, in which we allow users to vote for high quality keyphrases on papers' summary pages in CiteSeerX. These keyphrases are extracted using different models, but the model information is suppressed to reduce judgment bias. Voting systems are ubiquitous in social networks and multimedia websites, such as Facebook and YouTube, but they are rarely seen in scholarly digital libraries. A screenshot of an example of the voting interface is shown in Figure 6. A database is already setup to store the total number of counts for each voting type as well as each voting action. The database contains the following tables.

- **Model table.** This table contains information of keyphrase extraction models.
- **Voting table.** This table contains the counts of upvotes and downvotes of keyphrases extracted using all models from all papers. The table also records the time the voting of a keyphrase is last updated. The same keyphrase extracted by two distinct models will have two entries in this table.
- **Action table.** This table contains information of all voting actions on keyphrases, such as the action time, the type of action (upvote vs. downvote), the IDs of keyphrases voted, and the IDs of voters. A voter must log in first before they can vote. If a voter votes a keyphrase extracted by two models, two actions will be recorded in this table. If a user reverses his vote, two actions (unvote and vote) are recorded in this table.

The extraction modules can be evaluated by the summation of eligible votes over all papers. In classic supervised machine learning, predicted keyphrases are evaluated by comparing extraction results against the author-supplied keyphrases [10]. However, the list of author-supplied keyphrases may not be exhaustive,

i.e., certain pertinent keyphrases may be omitted by authors, but extracted by trained models. Crowd-sourcing provides an alternative approach that evaluates the pertinence of keyphrases from the readers' perspectives. However, there are certain potential biases that should be considered when deploying the system. One factor that can introduce bias is ordering because voters may not go through the whole list and vote all items. To mitigate this bias, we will shuffle keyphrases when displaying them on papers' summary pages. Another bias is the "Mathew's Effect" in which items with higher votes tend to receive more upvotes. We will hide the current votes of keyphrases to mitigate this effect.

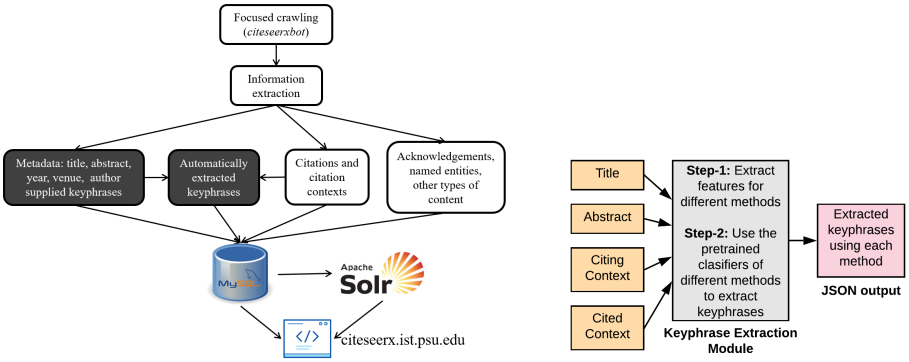
We plan to collect votes after opening the voting system for at least 6 months. Using this approach, the keyphrase extraction models can be evaluated at two levels. At the *keyphrase level*, we only consider keyphrases with at least 10 votes and apply a binary judgment for keyphrase quality. A keyphrase is "favored" if the number of upvotes is higher than the downvotes, otherwise, it is labeled as "disfavored". We can then score each model based on the number of favored vs. disfavored. At the *vote level*, we can score each model using upvotes and downvotes of all keyphrases. The final scores should be normalized by the number of keyphrase extracted by a certain model and voted by users.

6 Development and Deployment

Although CiteSeerX utilizes open source software packages, many core components are not directly available from open source repositories and require extensive programming and testing. The current CiteSeerX codebase inherited little from its predecessors (CiteSeer) for stability and consistency. The core part of the main web apps were written by Dr. Isaac Councill and Juan Pablo Fernández-Ramírez and many components were developed by other graduate students, postdocs and software engineers, which took at least 3-4 years.

CiteSeerX has been using keyphrases extracted using an unsupervised NP-Chunking method. This method is fast and achieves high recall, but it has a relatively low precision. Thus, we are exploring supervised models to extract keyphrases more accurately into CiteSeerX. Our keyphrase extraction module employs three methods: CeKE, Hulth, and KEA. The keyphrase extraction module runs on top of several dependencies, which handle metadata extraction from PDF files and document type classification in CiteSeerX. For example, GROBID [1] is used to extract titles, abstracts, and citing contexts. We also developed a program to extract cited contexts for a given article from the CiteSeerX database. In addition, a POS tagger[5] is a part of our keyphrase extraction module and is integrated in the keyphrase extraction module. Even though we selected CeKE-Citing, the keyphrase extraction package supports other variants of CeKE and it is straightforward to switch between them. Figure 7 shows the CiteSeerX system architecture and schematic diagram of our keyphrase extraction module.

[5] We have used NLP Stanford part of speech tagger.



(a) CiteSeerX architecture.

(b) Schematic diagram of keyphrase extraction module.

Fig. 7. CiteSeerX architecture and the keyphrase extraction module.

7 Maintenance

The keyphrase extraction module is developed and maintained by about 3 graduate students and a postdoctoral scholar in an academic setting. The keyphrase extraction project received partial financial support from the National Science Foundation. The maintenance work includes, but is not limited to fixing bugs, answering questions from GitHub users, updating extractors with improved algorithms, and rerunning new extractors on existing papers. Specific to the keyphrase extraction module, it can easily integrate new models trained on different or large data for the existing methods. In future, we aim to integrate new keyphrase extraction models. The key bottleneck is to integrate keyphrase modules into the ingestion system, so both author-supplied keyphrases and predicted keyphrases can be extracted with other types of content at scale. One solution is to encapsulate keyphrase extraction modules into Java package files (.jar files) or Python libraries so they can easily be invoked by PDFMEF [53], a customizable multi-processing metadata extraction framework for scientific documents. Currently, the CiteSeerX group is developing a new version of digital library framework that employs PDFMEF as part of the information extraction pipeline. The encapsulation solution can potentially reduce the maintenance cost and increase modularity.

8 Related Work

Both supervised and unsupervised methods have been developed for keyphrase extraction [25]. These methods generally consists of two phases. In the first phase, candidate words or phrases are extracted from the text using heuristics such as POS patterns for words or n-grams [28]. In the second phase, the candidate phrases are predicted as keyphrases or non-keyphrases, using both supervised and unsupervised approaches.

In the supervised studies, keyphrase extraction is formulated as a binary classification problem or a sequential labeling. In the binary classification, the candidate phrases are classified as either keyphrase or non-keyphrase. In the sequential labeling, each token in a paper (sequence) is labeled as part of a keyphrase or not [20, 40, 4]. The prediction is done based on different features extracted from the text of a document, e.g., a word or phrase POS tags, *tf-idf* scores, and position information, used in conjunction with machine learning classifiers such as Naïve Bayes, Support Vector Machines, and Conditional Random Field [28, 16, 38, 19]. The features extracted from external sources such as WordNet and Wikipedia [36, 34]; from the neighbourhood documents, e.g., a document's citation network [9, 8] were also used for the keyphrase extraction.

In unsupervised keyphrase extraction, the problem is usually formulated as a ranking problem. The phrases are scored using methods based on *tf-idf* and topic proportions [33, 6, 56]. The graph-based algorithms such as PageRank [37, 50, 21] and its variants [19, 15, 32] are also widely used in unsupervised models. Blank, Rokach, and Shani [7] ranked keyphrases for a target paper using keyphrases from the papers that are cited by the target paper and keyphrases from the papers that cite at least one paper that the target paper cites. The best performing model in SemEval 2010 [14] used term frequency thresholds to filter out unlikely phrases. Adar and Datta [3] extracted keyphrases by mining abbreviations from scientific literature and built a semantic hierarchical keyphrase database. Many of the above approaches, both supervised and unsupervised, are compared and analyzed in the ACL survey on keyphrase extraction by Hasan and Ng [26].

Usually, the performance of the supervised keyphrase extraction models is better than the unsupervised models [26].

9 Conclusions and Future Directions

By analyzing access logs of CiteSeerX in the past 3 years, we found that there are 3% of keyphrases common across all years, while there are many keyphrases which are only clicked during a particular year. In this application paper, we proposed to integrate three supervised keyphrase extraction models into CiteSeerX which are more robust than the previously used NP-Chunking method. To evaluate the keyphrase extraction methods from a user perspective, we implemented a voting system on papers' summary pages in CiteSeerX to vote on predicted phrases without showing the model information to reduce potential judgment bias from voters.

In the future, it would be interesting to integrate other keyphrase extraction models as well as other information extraction tools such as name-entity extraction tool to improve the user experience.

10 Acknowledgements

We thank the National Science Foundation (NSF) for support from grants CNS-1853919, IIS-1914575, and IIS-1813571, which supported this research. Any opin-

ions, findings, and conclusions expressed here are those of the authors and do not necessarily reflect the views of NSF. We also thank our anonymous reviewers for their constructive feedback.

References

1. Grobid. https://github.com/kermitt2/grobid (2008–2020)
2. Abu-Jbara, A., Radev, D.: Coherent citation-based summarization of scientific papers. In: ACL: HLT. pp. 500–509 (2011)
3. Adar, E., Datta, S.: Building a scientific concept hierarchy database (schbase). In: ACL. pp. 606–615 (2015)
4. Alzaidy, R., Caragea, C., Giles, C.L.: Bi-lstm-crf sequence labeling for keyphrase extraction from scholarly documents. In: WWW. pp. 2551–2557. ACM (2019)
5. Augenstein, I., Das, M., Riedel, S., Vikraman, L., McCallum, A.: Semeval 2017 task 10: Scienceie-extracting keyphrases and relations from scientific publications. arXiv preprint arXiv:1704.02853 (2017)
6. Barker, K., Cornacchia, N.: Using noun phrase heads to extract document keyphrases. In: Advances in Artificial Intelligence. pp. 40–52. Springer (2000)
7. Blank, I., Rokach, L., Shani, G.: Leveraging the citation graph to recommend keywords. In: RecSys. pp. 359–362 (2013)
8. Bulgarov, F., Caragea, C.: A comparison of supervised keyphrase extraction models. In: WWW. pp. 13–14 (2015)
9. Caragea, C., Bulgarov, F., Godea, A., Gollapalli, S.D.: Citation-enhanced keyphrase extraction from research papers: A supervised approach. In: EMNLP (2014)
10. Caragea, C., Bulgarov, F.A., Godea, A., Gollapalli, S.D.: Citation-enhanced keyphrase extraction from research papers: A supervised approach. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL. pp. 1435–1446 (2014), http://aclweb.org/anthology/D/D14/D14-1150.pdf
11. Caragea, C., Wu, J., Gollapalli, S.D., Giles, C.L.: Document type classification in online digital libraries. In: Twenty-Eighth IAAI Conference (2016)
12. Chen, H.H., Treeratpituk, P., Mitra, P., Giles, C.L.: Csseer: an expert recommendation system based on citeseerx. In: JCDL. pp. 381–382 (2013)
13. Councill, I., Giles, C.L., Kan, M.Y.: ParsCit: an open-source CRF reference string parsing package. In: LREC. vol. 8, pp. 661–667 (2008)
14. El-Beltagy, S.R., Rafea, A.: Kp-miner: Participation in semeval-2. In: SemEval. pp. 190–193 (2010)
15. Florescu, C., Caragea, C.: Positionrank: An unsupervised approach to keyphrase extraction from scholarly documents. In: ACL. pp. 1105–1115 (2017)
16. Frank, E., Paynter, G.W., Witten, I.H., Gutwin, C., Nevill-Manning, C.G.: Domain-specific keyphrase extraction. In: IJCAI. pp. 668–673 (1999)
17. Giles, C.L., Bollacker, K.D., Lawrence, S.: Citeseer: An automatic citation indexing system. In: JCDL. pp. 89–98 (1998)
18. Giles, C.L., Bollacker, K.D., Lawrence, S.: CiteSeer: An automatic citation indexing system. In: JCDL. pp. 89–98 (1998)
19. Gollapalli, S.D., Caragea, C.: Extracting keyphrases from research papers using citation networks. In: AAAI. pp. 1629–1635 (2014)

20. Gollapalli, S.D., Li, X.L., Yang, P.: Incorporating expert knowledge into keyphrase extraction. In: AAAI. pp. 3180–3187 (2017)
21. Grineva, M., Grinev, M., Lizorkin, D.: Extracting key terms from noisy and multitheme documents. In: WWW. pp. 661–670 (2009)
22. Hall, D., Jurafsky, D., Manning, C.D.: Studying the history of ideas using topic models. In: EMNLP. pp. 363–371 (2008)
23. Hammouda, K.M., Matute, D.N., Kamel, M.S.: Corephrase: Keyphrase extraction for document clustering. In: MLDM. pp. 265–274. Springer (2005)
24. Han, H., Giles, C.L., Manavoglu, E., Zha, H., Zhang, Z., Fox, E.A.: Automatic document metadata extraction using support vector machines. In: JCDL. pp. 37–48. IEEE (2003)
25. Hasan, K.S., Ng, V.: Conundrums in unsupervised keyphrase extraction: making sense of the state-of-the-art. In: COLING. pp. 365–373 (2010)
26. Hasan, K.S., Ng, V.: Automatic keyphrase extraction: A survey of the state of the art. In: ACL. pp. 1262–1273 (June 2014)
27. Hong, K., Jeon, H., Jeon, C.: Personalized research paper recommendation system using keyword extraction based on userprofile. In: Journal of Convergence Information Technology (JCIT) (2013)
28. Hulth, A.: Improved automatic keyword extraction given more linguistic knowledge. In: EMNLP (2003)
29. Jurgens, D., Kumar, S., Hoover, R., McFarland, D., Jurafsky, D.: Measuring the evolution of a scientific field through citation frames. TACL **6**, 391–406 (2018)
30. Khabsa, M., Giles, C.L.: The number of scholarly documents on the public web. PloS one **9**(5) (2014)
31. Larsen, P., Von Ins, M.: The rate of growth in scientific publication and the decline in coverage provided by science citation index. Scientometrics **84**(3), 575–603 (2010)
32. Liu, Z., Huang, W., Zheng, Y., Sun, M.: Automatic keyphrase extraction via topic decomposition. In: EMNLP. pp. 366–376 (2010)
33. Liu, Z., Li, P., Zheng, Y., Sun, M.: Clustering to find exemplar terms for keyphrase extraction. In: EMNLP. pp. 257–266 (2009)
34. Lopez, P., Romary, L.: Humb: Automatic key term extraction from scientific articles in grobid. In: SemEval. pp. 248–251 (2010)
35. Mahata, D., Kuriakose, J., Shah, R.R., Zimmermann, R.: Key2vec: Automatic ranked keyphrase extraction from scientific articles using phrase embeddings. In: NAACL. pp. 634–639 (2018)
36. Medelyan, O., Frank, E., Witten, I.H.: Human-competitive tagging using automatic keyphrase extraction. In: EMNLP. pp. 1318–1327 (2009)
37. Mihalcea, R., Tarau, P.: Textrank: Bringing order into texts. In: EMNLP (2004)
38. Nguyen, T.D., Kan, M.Y.: Keyphrase extraction in scientific publications. In: ICADL, pp. 317–326. Springer (2007)
39. Orduña-Malea, E., Ayllón, J.M., Martín-Martín, A., López-Cózar, E.D.: Methods for estimating the size of google scholar. Scientometrics **104**(3), 931–949 (2015)
40. Patel, K., Caragea, C.: Exploring word embeddings in crf-based keyphrase extraction from research papers. In: K-CAP. pp. 37–44. ACM (2019)
41. Qazvinian, V., Radev, D.R.: Scientific paper summarization using citation summary networks. In: COLING. pp. 689–696. Manchester, United Kingdom (2008)
42. Qazvinian, V., Radev, D.R., Özgür, A.: Citation summarization through keyphrase extraction. In: COLING. pp. 895–903 (2010)
43. Ritchie, A., Teufel, S., Robertson, S.: How to find better index terms through citations. In: CLIIR. pp. 25–32 (2006)

44. Sefid, A., Wu, J., Ge, A.C., Zhao, J., Liu, L., Caragea, C., Mitra, P., Giles, C.L.: Cleaning noisy and heterogeneous metadata for record linking across scholarly big datasets. In: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019. pp. 9601–9606 (2019)
45. Sinha, A., Shen, Z., Song, Y., Ma, H., Eide, D., Hsu, B.J., Wang, K.: An overview of microsoft academic service (mas) and applications. In: WWW. pp. 243–246 (2015)
46. Song, I.Y., Allen, R.B., Obradovic, Z., Song, M.: Keyphrase extraction-based query expansion in digital libraries. In: JCDL. pp. 202–209 (2006)
47. Tan, C., Card, D., Smith, N.A.: Friendships, rivalries, and trysts: Characterizing relations between ideas in texts. arXiv preprint arXiv:1704.07828 (2017)
48. Teregowda, P., Urgaonkar, B., Giles, C.L.: Cloud 2010. In: 2010 IEEE 3rd International Conference on Cloud Computing. pp. 115–122 (2010)
49. Treeratpituk, P., Giles, C.L.: Disambiguating authors in academic publications using random forests. In: JCDL. pp. 39–48. ACM (2009)
50. Wan, X., Xiao, J.: Single document keyphrase extraction using neighborhood knowledge. In: AAAI. vol. 8, pp. 855–860 (2008)
51. Williams, K., Wu, J., Choudhury, S.R., Khabsa, M., Giles, C.L.: Scholarly big data information extraction and integration in the citeseer digital library. IIWeb pp. 68–73 (2014)
52. Wu, J., Kandimalla, B., Rohatgi, S., Sefid, A., Mao, J., Giles, C.L.: Citeseerx-2018: A cleansed multidisciplinary scholarly big dataset. In: IEEE Big Data. pp. 5465–5467 (2018)
53. Wu, J., Killian, J., Yang, H., Williams, K., Choudhury, S.R., Tuarob, S., Caragea, C., Giles, C.L.: Pdfmef: A multi-entity knowledge extraction framework for scholarly documents and semantic search. In: K-CAP. pp. 13:1–13:8. ACM (2015)
54. Wu, J., Liang, C., Yang, H., Giles, C.L.: Citeseerx data: Semanticizing scholarly papers. In: SBD. pp. 2:1–2:6. ACM (2016)
55. Wu, J., Williams, K., Chen, H., Khabsa, M., Caragea, C., Ororbia, A., Jordan, D., Giles, C.L.: CiteSeerX: AI in a digital library search engine. In: AAAI. pp. 2930–2937 (2014)
56. Zhang, Y., Milios, E., Zincir-Heywood, N.: A comparative study on key phrase extraction methods in automatic web site summarization. JDIM **5**(5), 323 (2007)