Noname manuscript No.
(will be inserted by the editor)

Using Hybrid Algorithmic-Crowdsourcing Methods for Academic Knowledge Acquisition

Zhaoan Dong · Jiaheng Lu · Tok Wang Ling · Ju Fan · Yueguo Chen

the date of receipt and acceptance should be inserted later

Abstract Scientific literature contains a lot of meaningful objects such as Figures, Tables, Definitions, Algorithms, etc., which are called Knowledge Cells hereafter. An advanced academic search engine which could take advantage of Knowledge Cells and their various relationships to obtain more accurate search results is expected. Further, it's expected to provide a fine-grained search regarding to Knowledge Cells for deep-level information discovery and exploration. Therefore, it is important to identify and extract the Knowledge Cells and their various relationships which are often intrinsic and implicit in articles. With the exponential growth of scientific publications, discovery and acquisition of such useful academic knowledge impose some practical challenges For example, existing algorithmic methods can hardly extend to handle diverse layouts of journals, nor to scale up to process massive documents. As crowdsourcing has become a powerful paradigm for large scale problem-solving especially for tasks that are difficult for computers but easy for human, we consider the problem of academic knowledge discovery and acquisition as a crowd-sourced database problem and show a hybrid framework to integrate the accuracy of crowdsourcing workers and the speed of automatic algorithms. In this paper, we introduce our current system implementation, a **P**latform for **A**cademic k**N**owledge **D**iscovery and **A**cquisition (PANDA), as well as some interesting observations and promising future directions.

Keywords Knowledge acquisition · Crowdsourcing · Knowledge cells · Academic knowledge graph

1 Introduction

With an exponential growth of scientific publications, the wealth of academic knowledge within scientific publications is of significant importance for researchers. Traditional web-based systems usually provide literature search and retrieve services through a user-friendly search interface, such as Google scholar[1], DBLP[2], ACM Digital Library[3], arXiv[4], etc. They enable users to find papers via keywords or via faceted search on some metadata information including the *title*, *abstract*, *journal name* or *conference name*, information about the *authors* (e.g. *names*, *e-mails*, *affiliations*), and then rank the related papers according to the *relevance*, *citations* and *published date*, etc. There is no doubt that this kind of search pattern has brought great convenience in the last decade. However, researchers are often overwhelmed by the long list of search results. They have to scan the paper list and download some of them to read

Jiaheng Lu (🖂)
Department of Computer Science, University of Helsinki, Helsinki, Finland.
E-mail: jiahenglu@gmail.com

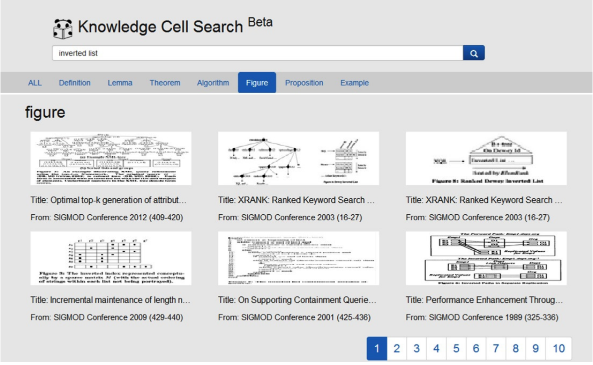
Zhaoan Dong · Jia heng Lu · Ju Fan · Yueguo Chen
DEKE, MOE and School of Information, Renmin University of China, Beijing, China.
E-mail: dongzhaoan@163.com

Tok Wang Ling
Department of Computer Science, School of Computing, National University of Singapore, Singapore.
E-mail: lingtw@comp.nus.edu.sg

Ju Fan
E-mail: fanju1984@gmail.com

Yueguo Chen
E-mail: chenyueguo@gmail.com

[1] http://scholar.google.com/
[2] http://dblp.uni-trier.de/
[3] http://dl.acm.org/
[4] http://arxiv.org/



Title: Optimal top-k generation of attribut...
From: SIGMOD Conference 2012 (409-420)

Title: XRANK: Ranked Keyword Search ...
From: SIGMOD Conference 2003 (16-27)

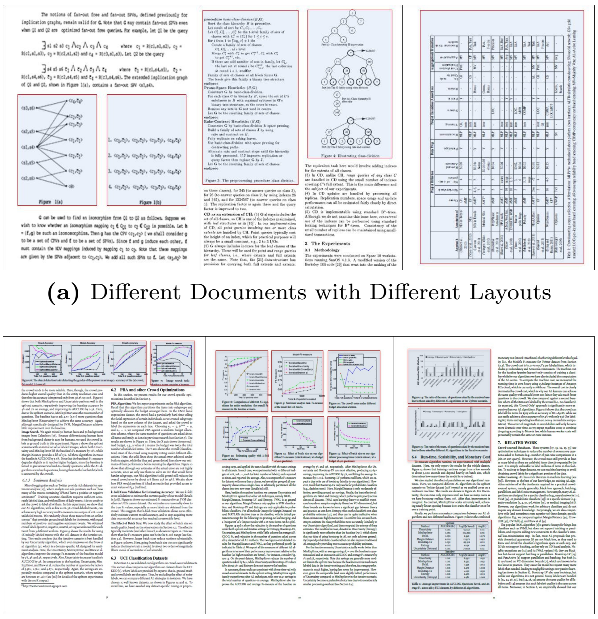
Title: XRANK: Ranked Keyword Search ...
From: SIGMOD Conference 2003 (16-27)

Title: Incremental maintenance of length n...
From: SIGMOD Conference 2009 (429-440)

Title: On Supporting Containment Querie...
From: SIGMOD Conference 2001 (425-436)

Title: Performance Enhancement Throug...
From: SIGMOD Conference 1989 (325-336)

Fig. 1 Knowledge Cell Search Results of Pandasearch



(a) Different Documents with Different Layouts

(b) Identical Document with Different Layouts

Fig. 2 An Example of Layouts of Figures

one by one. It is very time-consuming and costly especially when some papers are found useless and dropped by the users at last. Recently, we have developed a novel academic search engine named PandaSearch [11] that aims to provide a fine-grained academic search. As is shown in Fig. 1, when users submit a keyword like "*inverted list*", the system returns a list of meaningful information objects such as Definitions, Algorithms, Figures, Tables, and Theorems that are most relevant to the keyword, instead of a long list of papers. All of these meaningful objects will be defined as "**Knowledge Cells**" in Section 3, and what's more, Knowledge Cells and their various relationships could be used to build an "**Academic Knowledge Graph**" which would help to not only improve the results' quality of traditional academic search but also provide comprehensive information discovery and exploration. For example, an Algorithm A proposed in one article might have association with Algorithm B in another article, while it describes the relationship between Algorithm B and Definition C. Thus we can deduce the relationship A-C via B which may be never mentioned in both articles.

Obviously, the most important prerequisite is to correctly identify and extract Knowledge Cells including the names, contents and contexts, as well as some relationships among them. To achieve this goal, we face the following challenges.

The first challenge is to identify and extract each Knowledge Cell correctly. Although PDF has become the de facto standard of science literature, a PDF document is more complex than it seems. Human can easily deduce the structure and semantics of different characters and pictures on a page, but it is hard for computer algorithms. The main reason is that PDFs intrinsically do not contain or store enough structural information, they only provide the rendering information of individual text fragments for final presentation. Currently, a range of methods and techniques have been employed to identify the regions as chunks or blocks from PDFs and classify them into "rhetorical" categories through combinations of heuristics, rule-based methods, clustering and supervised learning [17]. However, they can hardly extend and scale up due to: (i) the variety of different journal layouts, and (ii) specific characteristics of each type of Knowledge Cells, as well as (iii) the layouts of Knowledge Cells often vary with different documents. For example, in Fig. 2a, the layouts of Knowledge Cells are always changing with different documents from different conferences or journals. In Fig. 2b, we selected page 9-11 from [25]. We can see that even in the same document, the layouts of Knowledge Cells are still different with each other. There are at least three different layouts of 11 logical objects including one Table and 10 Figures. Therefore, this poses a cumbersome task to current rule-based or machine learning based extracting algorithms.

The second challenge is to extract the contents, key phrases and contexts of Knowledge Cells to facilitate users in searching information. Sometimes, important information about Knowledge Cells is implied in the captions of Figures, specifications of Algorithms, and the content of a Knowledge Cell is hard for a computer to understand precisely. This poses a great challenge for algorithms to precisely extract the Knowledge Cells.

The third challenge is to extract the various semantic relationships between Knowledge Cells in order to build an Academic Knowledge Graph. As mentioned above, the relationships are usually implied or hidden in the sentences of the article. For example, if an article says "*We continue the example of Figure XXX to illustrate the algorithm of ...*", it usually indicates the relationship between a Figure and an Algorithm in the identical article. And another sentence, e.g. "*By Theorem YYY and Theorem ZZZ of [WWW], this theorem is proved...*"can be used to introduce the relationships of several Theorems from two different papers. Sometimes, the relationships tend to be rare and may not explicitly appear in any specific sentence. Moreover, some relationships require expertise to be recognized. Hence textual analytic techniques using Natural Language Processing or Machine Learning algorithms hardly return perfect results.

As *crowdsourcing* has become a powerful paradigm for large scale problem-solving especially for those tasks that are difficult for computers but easy for human [8, 24], we make use of crowdsourcing to identify and extract those Knowledge Cells as well as their relevant key information and relationships from huge amount of PDFs. It is notable that during the process of identification and extraction, some activities can generally be broken into small tasks which are often repetitive and do not require any specific expertise. For example, a human worker can almost effortlessly locate the content of a **Figure** by browsing the PDF pages and then crop the content only by "*drag and draw*". The cooperation of human and machine participants can help researchers to resolve large-scale complex problems in a more efficient way. On the one hand, leveraging human input can bring higher accuracy. On the other hand, if a great number of PDFs are crowdsourced, the cost will dramatically increase in terms of money or the processing time. Therefore, the natural alternative is to combine the accuracy of human with the speed and cost effectiveness of computer algorithms.

In summary, this article makes the following contributions:

- We stated the problem of academic knowledge discovery and acquisition as a crowd-sourced database problem where scholarly papers, Knowledge Cells and the relationship between Knowledge Cells are represented as rows or records with some missing attributes that could be supplied by either automatic algorithms or anonymous human workers.
- We proposed a hybrid framework integrating the accuracy of human workers and the efficiency of automatic algorithms to address the problem. We elaborated the academic knowledge graph in a two-level view, i.e., a graph which contains at least two types of nodes: paper nodes and Knowledge Cell nodes. These nodes are connected via various relationships, e.g., references of papers, the relationships between a Knowledge Cell and its papers, and the relationships between two Knowledge Cells (See Fig. 4). We implemented the system, a **P**latform for **A**cademic k**N**owledge **D**iscovery and **A**cquisition (PANDA), which consists of three main components: data acquisition, data storage and search. We developed an algorithm to filter away the PDF pages that do not contain target Knowledge Cells. In order to integrate the accuracy of human to improve the filtering performance, we extended the algorithm to a hybrid version which is adopted from the ideas of Uncertainty and MinExpError proposed by [25].
- In order to evaluate the effectiveness and the accuracy of anonymous workers, we conducted three crowdsourcing experiments on Amazon Mechanical Turk for PDF page filtering, boundary identification and review. In the page filtering tasks, the average accuracy of human workers achieves 92.8%, while identifying tasks achieves 93% and the review tasks 95%.

The remainder of this paper is organized as follows: In Section 2, we briefly overview the related work. Then, we give the definitions of Knowledge Cell and Academic Knowledge Graph as well as the statement of the problem (Section 3). Next, Section 4 gives an overview of academic knowledge acquisition framework, and Section 5 introduces the current system implementation and primary experiments results. Finally, Section 6 concludes the paper and gives insights into future work.

2 Related Work

In this section, several outstanding tools and systems that are most similar to our research are firstly reviewed and then we review the existing studies on information extraction from scientific literatures. We also introduce the state-of-the-art studies on crowdsourcing which will play crucial roles in our research work.

2.1 Management of Knowledge within Scientific Literature

Scientific literature contains some academic knowledge which is valuable but previously unknown. Tremendous interests have been given to extraction and management of research data within scientific literature.

One example is *Digital Curation* (DC)[5] which indicates the activities to maintain research data long-term including selection, preservation, maintenance, collection and archiving of digital assets and the process of

[5] https://en.wikipedia.org/wiki/Digital_curation

extraction of important information from scientific literature. Another example is *Deep Indexing*(DI)[6] which is used in ProQuest[7]to index the research data within scholarly articles that are often invisible to traditional bibliographic searches. In ScienceDirect[8], an advanced images search returns only figures, photos and video, not articles. Figures and tables are also listed in the left pane of the full-article page. And in CiteSeer$X$[9], it allows the users to search keywords and text within snippets in or around tables. However, unlike these systems that mainly focus on searching Figures and Tables, in PandaSearch [11], we aim to provide more extensive search over more diverse categories of Knowledge Cells, such as Definitions, Algorithms, Theorems, Lemmas, and so on.

There are also some other kinds of academic knowledge such as similarity relationships between scientific documents, trending articles in hot topics, etc. Most recently, for example, Alewiwi et al. [1] proposed an efficient filtering method based on the Z-order prefix to find highly similar documents. Swaraj et al. [31] proposed a fast approach to find trending articles and hot topics big bibliographic datasets. Note that our objective in this paper is to build an academic knowledge graph utilizing relationships of Knowledge Cells, which is significantly different with them.

2.2 Automatic Information Extraction from Scientific Literature

Along with the rapid expansion of digital libraries, PDF has been gradually a de facto standard of digital documents.

There are usually two ways to analyze and understand PDF documents, one of which is called *bottom-up* or *data-driven* method [10]. In these methods, the PDF pages are firstly converted into images and then rule-based information extracting techniques are performed. Identified characters are merged into words, words to sentences and then sentences to blocks, which would be classified into particular types (e.g. figure, caption, table, main text, title) using a combination of heuristics, clustering, and Machine Learning techniques. Geometrical relationships (e.g. rendering order and neighborhood) among these blocks are also utilized in the process [17]. Statistical methods and Artificial Intelligence techniques, including Probabilistic Modeling, Naïve Bayes Classifier and Conditional Random Field, Support Vector Machines are widely used [33]. Optical Character Recognition and Natural Language Processing techniques are also necessary for textual information extraction.

Another way is to directly analyze the PDF documents. Since the page model and document structure are already known in advance, these methods are named *model-driven* or *top-down* approaches [10]. Objects can be extracted directly by analyzing the layouts and page attributes (e.g. point size and font name). Here, many commercial or open-source tools such as PDFBox [10] and libSVM[11] can be exploited.

Nevertheless, as mentioned in Section 1, it is really a cumbersome task for current rule-based or machine learning based extracting algorithms to handle diverse layouts of journals, different characteristics of each type of Knowledge Cells, etc.

2.3 Task-Oriented Crowdsourcing

During the last decade, crowdsourcing has become popular among companies, institutions and universities as a promising on-line problem solving paradigm tapping the intelligence of the crowd. Crowdsourcing platforms such as Mechanical Turk have been widely applied to solve various tasks such as data collection, image labeling, recognition and categorization, translation [24], etc. Basically, the studies on crowdsourcing mainly focus on: (1) definitions and taxonomy; (2) applications and systems; (3) motivations and incentives; (4) task designing and assignment; (5) answers aggregation and quality control. All of the above aspects are thoroughly discussed in recent surveys [5, 9, 16, 24, 27, 30, 34, 35]. In computer science, crowdsourcing is highly connected to human computation [7, 19, 27], which replaces machines with humans in certain computational steps where humans usually perform better. Just as stated in [27] that crowdsourcing is a form of collective intelligence that overlaps human computation. In this subsection, we just briefly review recent progresses on task-oriented crowdsourcing such as task design, answers aggregation and quality control that are most relevant to our research.

2.3.1 Crowdsourcing task and workflow

According to [24], crowdsourcing tasks can be categorized into two types: *micro-tasks* and *complex-tasks*. While *micro-tasks* are atomic operations, *complex-tasks*

[6] http://proquest.libguides.com/deepindexing
[7] http://search.proquest.com
[8] http://www.sciencedirect.com/science/search
[9] http://citeseer.ist.psu.edu/
[10] http://pdfbox.apache.org/
[11] https://www.csie.ntu.edu.tw/~cjlin/libsvm/

are organized sets (e.g. *workflows*) of micro-tasks with a specific purpose.

When solving complex tasks, different methods, for example, *crash* and *rerun*,*map reduce*,*divide* and *conquer* can be utilized to manage workflows of tasks [24]. Some of the predefined templates or design patterns for task design, workflow design, and reviewing methodologies have been provided [3]. Sabou et al. [29] proposed a set of best practice guidelines for crowdsourcing task design. Luz et al. [23] proposed a semi-automatic workflow generation process for human-computer micro-task workflows. This process is based in a 3-layered architecture that defines the set of operations performed by micro-tasks on top of domain ontologies. Lofi et al. [22]extensively investigated hybrid crowdsourcing human computation workflows and abstracted generic design patterns. Each design pattern is described and discussed with a special focus on its requirements, constraints, and effects on the overall workflow.

2.3.2 Answers Aggregation

One of the biggest challenges of crowdsourcing is aggregating the answers collected from the crowd. On one hand, a number of human workers with different background or wide-ranging levels of expertise might lead to high contradiction and uncertainty. On the other hand, human workers are prone to error because of the carelessness, insufficient expertise or the difficulty of questions themselves. Additionally, malicious workers or spammers can submit random answers to pursuit monetary profit or rewards. Many aggregation techniques have been proposed, which are generally performed in two ways: *Non-Iterative* and *Iterative*. Majority Decision(MD) [20], for example, is a simple non-iterative approach that selects the answer with highest votes as the final value. While in iterative methods, such as Expectation and Maximization (EM) [13], a series of iterations will be performed. Each iteration contains two steps [12]: (1) update the aggregated value of each question based on the workers expertise, and (2) adjust the expertise of each worker based on the answers. The authors of [12] presented a benchmark to evaluate the performance of star-of-the-art aggregation techniques within a common framework. The metrics include *computation time*, *accuracy*, *robustness* and *adaptivity to multi-labeling*.

2.3.3 Quality control

A central challenge of crowdsourcing is how to keep balance between the expected monetary costs and results quality in mind. So far, some good mechanisms have been proposed to detect malicious behavior and fraud. For example, Rzeszotarski et al. [28] presented *CrowdScape*, a system that supports the human evaluation of complex tasks through interactive visualization and mixed initiative machine learning. Joglekar et al. [14]devised techniques to generate confidence intervals for worker error estimates. Allahbakhsh et al. [2] proposed a general framework for characterizing two main dimensions of quality control:worker profiles and task design. Dai et al. [4] and Panos et al. [26] separately devoted themselves to analyzing and optimizing existing workflows to improve both the quality and the cost of crowdsourcing. Li et al. [21] put forward a crowdsourcing fraud detection method to find out the spammer according to the psychological difference. Wang et al. [32] developed a machine learning model against practical adversarial attacks in the context of detecting malicious crowdsourcing activity.

2.4 Crowdsourcing as a Tool for Knowledge Acquisition

Crowdsourcing is a relatively new approach for knowledge acquisition. Based on this approach, many kinds of problems can be distributed and resolved through the adoption of appropriate web-based platforms. For example, Kamar [15] studied how to fuse human and machine contributions to predict the behaviors of workers and presented a principled approach for consensus crowdsourcing. Lofi et al. [22] extensively investigated hybrid crowdsourcing human computation workflows and abstracted five generic design patterns, such as *Magic Filter*, *Crowd Trainer*, *Machine Inprovement*, *Virtual Worker* and *High Confidence Switching*. Each pattern is described and discussed with a special focus on its requirements, constraints, and effects on the overall workflow and can be extended and combined to support more complex workflows. Kondreddi [18] presented Higgins, a novel system architecture that effectively integrates an automatic Information Extraction (IE) engine and a Human Computing (HC) engine. With the help of semantic resources like WordNet, ConceptNet, Higgins is used for knowledge acquisition by crowdsourced gathering of relationships between characters in narrative descriptions of movies and books. Mozafari et al. [25] proposed two Active Learning algorithms for labeling tasks in crowd-sourced databases, MinExpError and Uncertainty, to decide which items should be switched to the crowd. They also developed a crowdsourcing allocating technique, called Partitioning-Based Allocation (PBA), which dynamically partitions the unlabeled items according to difficulty and adjust the number of required human workers.

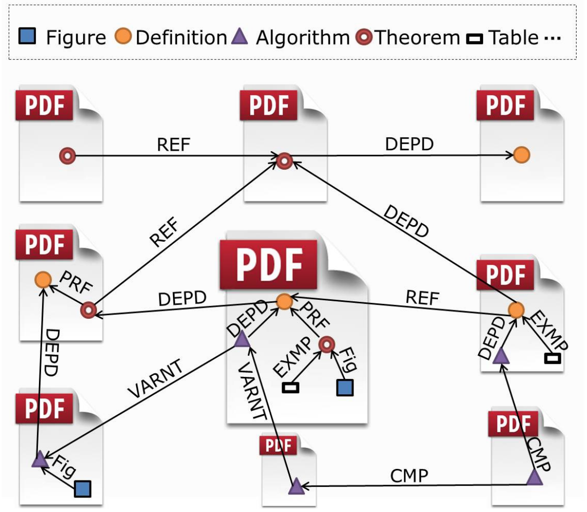


Fig. 3 A Fragment of an Academic Knowledge Graph

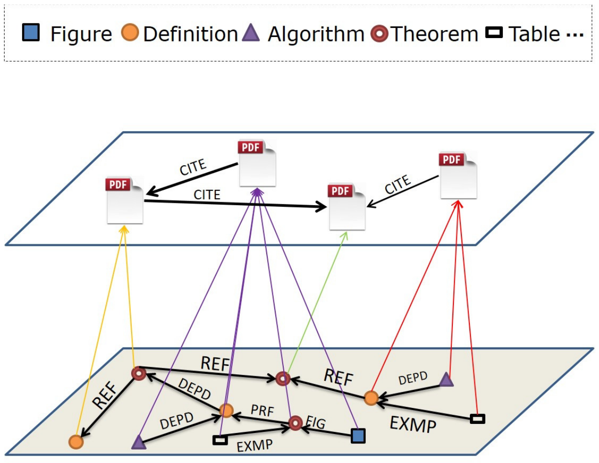


Fig. 4 A Two-Level View of a General Academic Knowledge Graph

Although there are already so many techniques and systems for knowledge acquisition, most of them are optimized for specific application domains or particular types of information and hence not well-suited for all kinds of information extraction tasks. They cannot be directly applied to academic knowledge discovery and acquisition with the consideration of challenges mentioned in Section 1.

3 Problem Statement

In this section, we first give the general definitions of Knowledge Cell and Academic Knowledge Graph.

Definition 1 ***A Knowledge Cell*** is a meaningful information object within an academic document. Each Knowledge Cell should have some attributes including an identifier (e.g. *kid*), paper identifier (e.g. *pid* that indicates the paper which this Knowledge Cell belongs to), type (e.g. *Definition*, *Figure*, *Theorem*, *Algorithm*, *Table*, *Lemma*, etc.), name (e.g. *algorithm name*, *definition name*, *figure caption*, *table caption*, etc.), content (e.g. the pseudo code of an Algorithm, the graphical area of a Figure, etc.) and key phrases (i.e. the reference contexts of a Knowledge Cell which are usually some sentences or paragraphs). Especially, papers are also of a special kind of Knowledge Cells that have attributes like paper identifier (e.g. *pid*), *title*, *authors*, *pages*, *conference* or *journal*, *date*, etc.

Definition 2 ***An Academic Knowledge Graph*** is a directed graph $AKG{=}(K, R)$, where K is the set of Knowledge Cells extracted from a collection of academic documents, and $R = \{(k_1, k_2, r) | k_1, k_2 \in K, k_1 \neq k_2$, and r is the relationship between k_1 and $k_2\}$. Note that k_1 and k_2 are two Knowledge Cells either from one PDF file or two different files.

For example, Fig. 3 illustrates a fragment of an Academic Knowledge Graph. We use different shapes to represent different kind of Knowledge Cells and arrows with different labels to represent various relationships. If the citation relationships between papers are also taken into account, we will obtain a General Academic Knowledge Graph (GAKG) which contains at least two categories of nodes and three kinds of edges (See Fig. 4.). One category of nodes are important papers and others are Knowledge Cells from those papers. The edges include the relationships between (i) each paper to its citations, (ii) each Knowledge Cell to its paper, and (iii) two relevant Knowledge Cells. Thus the General Academic Knowledge Graph could be viewed in a two-level manner. The top level is at paper level and about all the references between papers. The "**deep level**" is of Knowledge Cells with their detailed relationships. These two levels could be navigated from each other via the edges between Knowledge Cells and their papers.

With the aid of an Academic Knowledge Graph, academic search engines could provide more accurate search results for a deep-level information discovery and exploration. For example, they could improve the ranks of papers that contain those Knowledge Cells matching the keywords. And they could also provide a fine-grained search regarding to Knowledge Cells directly, such as Definitions, Algorithms, Figures, Tables, etc. By this way, we can look "**inside**" the PDF documents instead of scanning the long list of papers. Further, academic search engines could provide a set of SQL-like

APIs for developers and external systems as demonstrated in the following examples. [12].

Example 1 : Consider a query to find the Figures that contain keywords "*inverted list*" in their captions. At the same time, we also want to get the titles of papers in which those Figures appear.

```
SELECT p.pid, p.title, k.name, k.content
FROM   papers p, cells k
WHERE  contains(k.name,"inverted list")
       AND k.type="Figure"
       AND p.pid=k.pid;
```

To support this query, the search engine should find **Figures** from Knowledge Cells that contain "*inverted list*" in their captions. Unless the Figures have been previously obtained and stored in a repository, we must identify and extract them by automatic algorithms or soliciting human workers. Additionally, we need to extract the name, caption, content and other attributes of each Knowledge Cell for more queries. If some values of these attributes are missing, automatic algorithms or human workers will be invoked to fill them.

Example 2 : To find those algorithms which are variants or have been compared with an **Algorithm** whose name is related to "*hash join*" algorithm. Especially, we hope that the two Algorithms mentioned above are from different papers.

```
SELECT k1.pid, k1.name, k2.pid, k2.name
FROM   cells k1, cells k2
WHERE  relations(k1,k2) IN ("CMP","VARNT")
       AND contains(k2.name,"hash join")
       AND k1.type = k2.type = "Algorithm"
       AND k1.pid != k2.pid;
```

In Example 2, we assume that the relationships between k_1 and k_2 have been identified and extracted and represented in an Academic Knowledge Graph, where **CMP** can represent the *comparison* relationship between k_1 and k_2 and **VARNT** means k_1 is a *variant* or *extension* of k_2, for example, as is shown in Fig. 3 and Fig. 4.

More relationships between two arbitrary Knowledge Cells A and B (e.g. **REF** indicates A is referenced as B in another paper; **PRF** indicates A is referenced in proof of B; **DEPD** indicates A depends on B; **EXMP** indicates A is an example of B, etc.) can be manually identified and extracted by human workers with some hints/guidances or automatically by heuristic rules in the future extraction process. As described above, we can state the problem as follows.

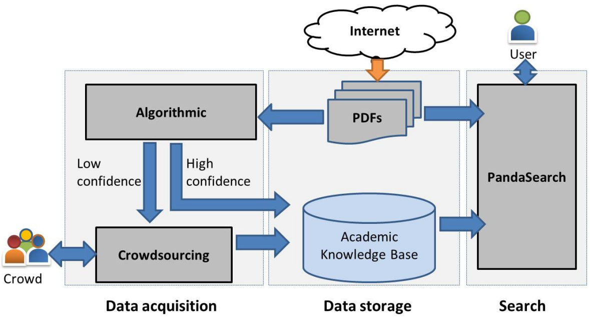


Fig. 5 The Architecture of the Platform

Problem Statement. In this research, the problem of academic knowledge discovery and acquisition can be modeled as a crowd-sourced database problem [6], where scholarly papers as well as the Knowledge Cells and their key phrases and relationships can be represented as rows or records with some missing attributes that could be supplied by either automatic algorithms or anonymous human workers. We mainly focus on how to design such hybrid workflows that could transparently combine the automatic algorithms and crowdsourced tasks.

4 An Overview of the Academic Knowledge Acquisition Framework

Our platform, named PANDA (the Platform for Academic kNowledge Discovery and Acquisition) is integrated with our previous work on PandaSearch [11]. Our ultimate objective is to build a platform for researchers to find the desirable information within the scientific literature and to assimilate the research data quickly and effectively. Fig. 5 shows the architecture of the platform, which consists of three main components: **Data acquisition**, **Data storage** and **Search**. In this paper, we mainly focus on the component of data acquisition. We propose a hybrid framework for academic knowledge discovery and acquisition, i.e, identifying and extracting Knowledge Cells and their relationships from PDF documents. We briefly describe the framework as a multi-stage process as follows.

(1) Preprocessing stage. In this stage, we collect a large corpus of PDF documents by crawling public websites. Metadata information of each paper including the *title*, *abstract*, *journal name* or *conference name*, information about the *authors* also should be harvested in

[12] We extend the standard SQL statements to illustrate these examples. Tables like *papers* and *cells* can be either relational tables or non-relational data collections, and functions like "relations" and "contains" can be some built-in functions. It doesn't affect the problem statement.

advance from DBLP, Google Scholar, etc. Next, in order to perform text analysis for extracting the topics and contexts of each Knowledge Cell, they should be firstly converted into a standard textual format. Further, it is necessary to split each PDF document into pages for automatic extraction and Human Intelligence Tasks. Some PDF pages that obviously do not contain the target Knowledge Cells should be filtered away.

(2) Extracting knowledge using automatic algorithms. In this stage, heuristic methods and machine learning algorithms are employed to identify and extract Knowledge Cells and their relationships. In our hybrid framework, they should also provide a confidence estimation on how accurate and reliable an identified result is likely to be. According to the confidence value, the results with high value will be retained. Otherwise, the current page will be switched to the crowdsourcing layer as a Human Intelligence Task Candidate (HITC). Obviously, special strategies have to be designed to make the algorithms confidence-aware, i.e., transmitting the extracting tasks with low confidence to the crowdsourcing platform, otherwise accepting the results. The most challenging work is how to define and calculate the confidence value and adjust the filtering threshold dynamically with consideration of time cost, result quality and budget of crowdsourcing.

(3) Designing crowdsourcing tasks. In this stage, Human Intelligence Tasks (HITs) for extracting certain Knowledge Cell will be generated based on the set of Human Intelligence Task Candidates. Based on the hybrid algorithmic-crowdsourcing work-flows, we aim to build a task-oriented crowdsourcing system. Human workers would be recruited for generating initial training dataset or manually confirming the ambiguous results for the algorithmic peer. Various tasks including identifying Knowledge Cells, reviewing other worker's answers are published through web-based interfaces.

(4) Crowdsourcing process management. While undertaking the crowdsourcing tasks, human workers may make innocent or deliberate mistakes. Crowdsourcing answers aggregation and quality control issues will be investigated to guarantee the quality of results. From the perspective of quality control, we should develop a tutorial module and a test module. Human workers have to take the tutorial tasks to learn how to perform the tasks and pass the test, otherwise, they could not apply the formal extraction tasks and review tasks. A crowdsourcing cost model is also crucial for our research. We try to study how to achieve a higher quality with a fixed budget, or complimentary, how to reduce the cost with quality constraints.

5 Current System Implementation

In this section, we introduce the system implementation and interesting experimental results. The architecture of current system implementation is divided into 4 layers, as is shown in Fig. 6.

5.1 Data Storage

There are mainly two data stores (See Fig. 6.). One is the PDF documents repository. More than 2.9 Million PDF documents have been crawled from the public websites. The other important part of data store is the Academic Knowledge Base, where the extracted Knowledge Cells and the Academic Knowledge Graph are stored. We list the data type and the corresponding number we have obtained in Table 1. The current volume of the whole dataset is nearly 4 Tera bytes.

5.2 Algorithmic Layer

Currently, we have built an algorithm to filter away the PDF pages that do not contain target Knowledge Cells. Considering the sparsity of each kind of Knowledge Cells, the page filtering algorithm is obviously helpful to reduce the workloads of identification of Knowledge Cells. For example, we observed that nearly 70% PDF pages do not contain Figures in our sampled 723 PDF documents. Of course, we should first provide a specification to tell which kind of Knowledge Cells to be identified and extracted.

In this paper, we mainly focus on page filtering algorithm for Figures, leaving other categories of Knowledge Cells for future work. Specifically, given a set of unlabeled PDF pages, our filtering algorithm should firstly label each of them whether it contains Figures or not. It is actually a simple binary classifier which is first trained based on the extracted features from the labeled PDF pages. The extracted features, for example, include: (1) Whether it contains a new line beginning with keywords "Figure" or "Fig." or "FIGURE"; (2) The keyword is followed by a number; (3) Bold font or not; (4) Capitalized or not, (5) The number of rows in the page; (6) The average length and width of text rows in this page, (7) Is it a "Title page", (8) Is it a "cover page" etc.

The fundamental challenge here is to design a selection strategies (e.g., a score function that returns a confidence value) that takes the difficulty of identifying a Knowledge Cell into consideration. Currently, we adopt the ideas of Uncertainty and MinExpError proposed by Mozafari, et.al [25]. Uncertainty algorithm

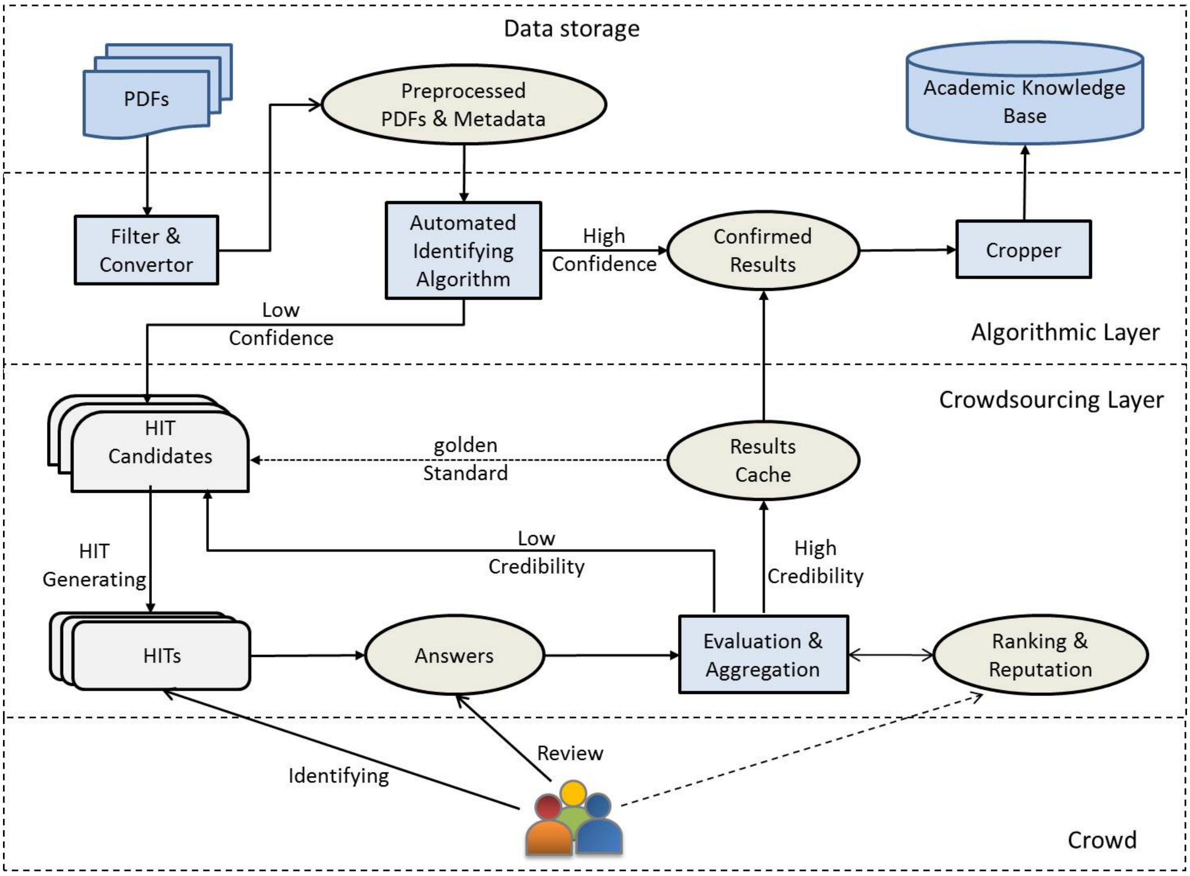


Fig. 6 The Implementation of the Prototype

Table 1 Statistics of Current Data Stores

Data Type	Number
Papers	2975828
Figures	25427
Tables	14392
Definitions	8934
Lemmas	757
Theorems	1026
Algorithms	1371
Propositions	52
Examples	1038

aims to ask the crowd to label the PDF pages that are difficult for the classifier but relatively easy for human workers. Intuitively, the difficulty refers to the uncertainty of a classifier, that is to say, the more uncertain the classifier is the more likely it will mislabel the page. But as observed in [25], those pages that have different labels with the classifier results may have larger impact on the classifiers performance. Thus MinExpError algorithm takes into account both the uncertainty and the false labeled data items. That is, we aim to assign the most beneficial PDF pages to human workers under a given crowdsourcing budget, for example, the number of Human Intelligence Tasks. The binary classifier operates in the straightforward manner. Because the PDF documents naturally can be partitioned into different groups according to their publishing venues or years. The PDF pages in a same group usually have similar document structures and layouts, so it is easy for the pageFilter to process. What we should do is provide some labeled items as initial training data for each group. The number of crowdsourced PDF pages k is determined by two parameters: an expected F-measure Q and an expected maximum of HITs N.

The Uncertainty and MinExpError algorithms are detailed in [25], so we just give the pseudocode of the pageFilter as is shown in Algorithm 1. In our current settings, the pageFilter takes as input (i) a set of unlabeled PDF pages U; (ii) a set of labeled PDF pages T as training set; (iii) a score/ranking function F which operates based on a binary classifier C; (iv) a switch strategy S, and (v) a budget B, i.e, an expected maximum of HITs N and an F-Mesure value Q. For example, we evaluate the pageFilter over 723 PDF documents

Algorithm 1 *pageFilter*

```
Input:
    T is a set of labeled PDF pages.
    U is a set of unlabeled PDF pages.
    F is a score/ranking function.
    S is a switch strategy.
    B is a budget, i.e, a maximum N of HITs and an F-Mesure value Q.
Output:
    R is a set of PDF pages that contain Figures, $R \subset U$.
Compute the scores for pages in U: $W \leftarrow F(C,T,U)$;
Select $U' \subset U$ based on $S(W,B(N,Q),U)$;
Send the PDF pages in U-U' to the binary classifier C
    for labels, i.e., $MR \leftarrow C(T,U-U')$;
Send the pages in U' to crowd: $CR \leftarrow Crowd(U')$;
$R \leftarrow MR \cup CR$;
return R;
```

Algorithm 2 *boundaryDetector.*

```
Input: A set of PDF pages, D.
Output: The locations of Figures in PDF pages, R.
PDFpages $\leftarrow$ spliter(D);
TextFile $\leftarrow$ convertor(D);
FilteredPages $\leftarrow$ pageFilter(PDFpages);
while FilteredPages.hasMore() do
    CurPage $\leftarrow$ FilteredPages.nextPage();
    Locations $\leftarrow$ Rule-based-Locating(CurPage, TextFile);
    while Locations.hasMore() do
        curPostion $\leftarrow$ Locations.nextPosition();
        (UpLeftX, UpLeftY, LowRightX, LowRightY)
            $\leftarrow$ Boundary_Detector(CurPage, curPosition);
        $R \leftarrow R \cup$ (UpLeftX, UpLeftY, LowRightX, LowRightY);
    end while
    return R;
end while
```

containing 5514 PDF pages. We labeled 500 pages as training set, 293 of them contain Figures and 207 pages without any Figures. We set the F-Measure to 90% and the maximum of HITs to $N = 551$ (i.e., 10% of the unlabeled PDF pages). At last, we obtained $k = 473$ PDF pages switched to the crowdsourcing layer, otherwise, $k = 551$ pages will be crowdsourced.

We also have built a boundary detector to identify the location and the boundary rectangle of each Figure. As is shown in Algorithm 2, the first step is to split the PDF document into pages. And then call the pageFilter to filter the pages that do not contain Figures. The locations of Figures are found by locating their captions in the paper. To identify the captions, we analyze the texts and layout of the page converted by PDFBox. We also take advantage of the open source libSVM classifier to identify the bounding rectangles of Figures based on the bounding boxes of all the text blocks, the fonts and font sizes, the height of lines, etc. The up-left and low-right corners of a bounding rectangle are computed by an algorithm, and then sent to an image cropper (i.e. the Cropper in Fig. 6.) for segmenting. The cropped image is indexed and stored for further usage.

We perform an initial experiment for extracting Figures within nearly 4,000 SIGMOD papers from 1980 to 2014. To evaluate the performance of boundary detector, we use ***Completeness*** and ***Purity*** in addition to the common metrics in IR: ***Precision***, ***Recall*** and ***F-Measure.*** A Knowledge Cell's graphical component is ***complete*** when it includes all the objects in the exact region and ***pure*** if it does not contain anything that does not belong to the Knowledge Cell. A correctly identified component of a Knowledge Cell is therefore both complete and pure. As an example, we give the definitions for evaluation measures of Figures as follows:

$$Recall = \frac{\#correctly\ identified\ Figures}{\#Figures\ in\ the\ paper}$$

$$Precision = \frac{\#correctly\ identified\ Figures}{\#identified\ Figures}$$

$$F\text{-}Measure = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

Fig. 7 shows the performance of current automatic algorithms for extracting Figures. The PDF files in the early years are scanned image files of the hardcopies of papers, which makes them difficult to be identified due to the low quality or resolutions. This is why the performance for papers from 1980 to 1989 are lower than those of the later years.

As can be seen in Table 1, the number of Figures is much more than other Knowledge Cells. This is because we currently focus on the extraction of Figures. The algorithms for extracting other Knowledge Cells, currently achieving 78% precision, 72% recall for Definitions and 84% precision, 75% recall for Algorithms [11] for average, are still under development and need to be further optimized. Hence we do not describe them here due to the space limitation.

5.3 Crowdsourcing Layer

Once algorithmic layer decides which PDF pages can be sent to the crowd, crowdsourcing tasks would be generated by the HIT generator. After being finished by human workers, the answers of HITs are aggregated and the workers are evaluated based on their performance. Answers with high credibility will be passed and directly output to results cache, otherwise rejected. The results in the cache will be moved to local storage, while the ranking and reputation of workers can be referenced by coming applications like task assignment.

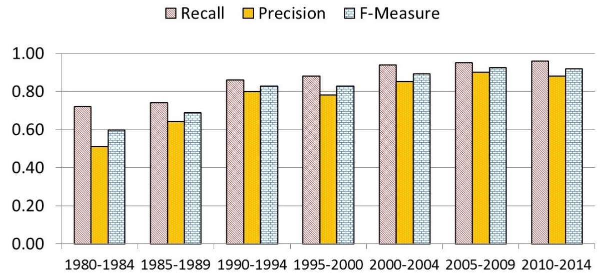


Fig. 7 Performance of Automatic Extracting Figures

Currently, we have developed several basic crowdsourcing workflows in the prototype system. There are mainly three basic Human Intelligence Tasks (HITs) including **Page Filtering**, **Boundary Identification** and **Review**. In Page Filtering, instead of simply asking the workers to confirm whether a PDF page contain the target Knowledge Cell or not, we ask the workers to find how many Figures in the PDF page. We group the PDF pages according to the number of Figures in each one and allocate them according to the group in the next two tasks. Identifying tasks ask workers to identify the Knowledge Cells from PDF pages. Review tasks ask workers to check the answers of other workers for the sake of quality control. Below we introduce the **Boundary Identification** and **Review** tasks in detail.

(1) Boundary Identification: As shown in Fig. 8a, the worker can click the "Crop " button and select the bounding boxes of Figures by "drag and draw". At the same time, the worker is also asked to input the sequential number of the Figures. And optionally, they can input the Caption of the Figure which are sometimes too difficult to be extracted by algorithms. At last, the results for current page can be saved by clicking the "Save" button. All the operations must be finished within a time limit, for example, 10 minutes. In order to keep the workers being active, tasks assigned to each worker should not be too many. We allocate 10 tasks to each worker in the example.

(2) Review: The goal of review tasks is to evaluate the answers contributed by other workers. For example, we ask one worker to crop the Figures from one PDF page, and send the answer to three reviewers to approve or reject. Each reviewer accepts or rejects the answer depending on his judgement of whether the image of a Knowledge Cell has been well segmented. We simply use Majority Vote of three reviewers at most for each review task. An answer will be passed if it is accepted by both of the two reviewers or rejected if both of them disagree. If two reviewers have different opinions, a third reviewer would be involved in and give a final result. This basic review method can evaluate answers with lower crowdsourcing cost because the third reviewer is not always invoked, especially when the tasks are easy enough for human workers to make a decision but too difficult for computer algorithms.

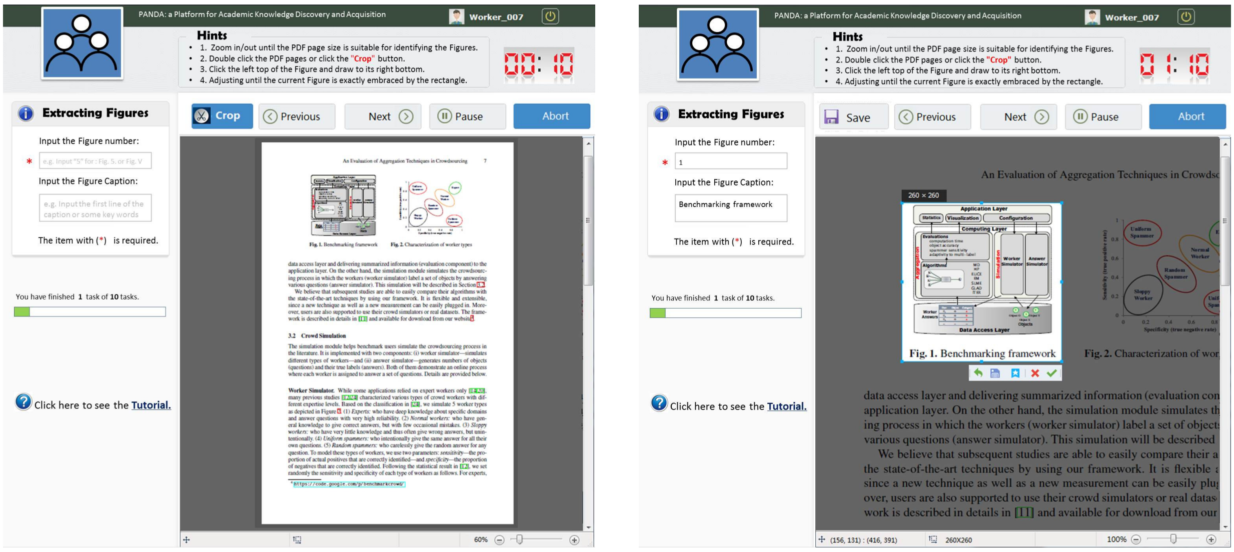
Finally, we send the confirmed data to our local server in terms of JSON which includes the page size, the final cropped positions and size as well as some descriptions of the Knowledge Cells. The Cropper of local server, as is shown in Fig. 6, will cutout and save the images according to the coordinateness.

5.4 Experiments on Amazon Mechanical Turk

A crucial issue of our hybrid framework is how to recruit enough workers without special training and expertise to correctly identify and extract Knowledge Cells. To address this problem, as mentioned above, we try to break the crowdsourcing tasks into several small and simple tasks which do not require any specific expertise. In order to evaluate the effectiveness and the accuracy of anonymous human workers, we perform 3 experiments on Amazon Mechanical Turk[13]: **Page Filtering**, **Boundary Identification** and **Review**. We published the 473 PDF pages switched to the crowdsourcing layer. We firstly recruit some student volunteers to labeled them for an initial training set. For Page Filtering tasks, the average accuracy of AMT workers achieves 92.8%, while Boundary Identification achieves 93% and Review achieves 95%.

Another important issue is the huge amount of PDF documents, that is, it is unfeasible to recruit enough human workers on crowdsourcing platform like Amazon Mechanical Turk to finish the work. To overcome such scaling problems, we could embed the crowdsourcing tasks into our PandaSearch system and engage the users of the system help us to identify Knowledge Cells. For example, in the future, we can allow authors to upload their published papers to our system and do the crowdsourcing tasks. To this end, we design and implement a user-friendly interface. We can also invite the users of PandaSearch in the feedback loops. As illustrated in Fig. 4, the returned results towards a query can be also organized in a two-level manner. When users click a paper in the list, the system could display all the Knowledge Cells of the paper. While reading the content of each Knowledge Cell, the users could give feedback by confirming the information of the Knowledge

[13] https://s3-us-west-2.amazonaws.com/cropfigure/templates.html



PANDA: a Platform for Academic Knowledge Discovery and Acquisition

Hints
- 1. Zoom in/out until the PDF page size is suitable for identifying the Figures.
- 2. Double click the PDF pages or click the "Crop" button.
- 3. Click the left top of the Figure and draw to its right bottom.
- 4. Adjusting until the current Figure is exactly embraced by the rectangle.

Extracting Figures

(a) Input the Figure number: e.g. Input "5" for : Fig. 5 or Fig V

Input the Figure Caption: e.g. Input the first line of the caption or some key words

(b) Input the Figure number: 1

Input the Figure Caption: Benchmarking framework

The item with (*) is required.

You have finished 1 task of 10 tasks.

Click here to see the Tutorial.

(a) Before Identifying the Boundaries **(b)** After Identifying the Boundaries

Fig. 8 An Example of Web-based Interfaces for Extracting Figures

Cell or reporting errors they find. Further, they could correct the errors via predefined interfaces.

6 Conclusion and Future Plan

In this paper, we stated the problem of academic knowledge discovery and acquisition within the scientific literature as a crowd-sourced database problem where scholarly papers, Knowledge Cells and the relationships between Knowledge Cells are represented as rows or records with some missing attributes that could be supplied by either automatic algorithms or anonymous human workers. We proposed a hybrid framework which integrates the accuracy of human workers and the speed of automatic algorithms to address the problem and described our current system implementation, a **P**latform for **A**cademic k**N**owledge **D**iscovery and **A**cquisition (PANDA).

In the future, we firstly plan to improve the feasibility of the crowdsourcing interfaces and optimize the design of HITs. Secondly, we will enhance current algorithms with the capabilities of confidence-aware and iterative interaction with the crowdsourcing module. Specifically, it can be realized based on the following aspects: **(1) Selection strategies** which can be used to choose pages that will be sent to human workers; **(2) Optimization** for the performance of automatic algorithms with the aid of human contributions. For example, crowd can provide training data or help to validate the ambiguous answers; **(3) Trade-off considerations** about achieving a higher quality within a given budget, or reducing the whole cost in terms of time and money with quality constraints. Finally, we will extend the framework to other Knowledge Cells, as well as the relationships among them to finally construct the Academic Knowledge Graph. Our ultimate goal is building a system for researchers to find the desirable information within the scientific literature and to assimilate the research data quickly and effectively.

Acknowledgements This study was funded by the National Natural Science Foundation of China (Grant No.61472427) and the Research Funds of Renmin University of China (Grant No.11XNJ003).

References

1. Alewiwi, M., Orencik, C., Savaş, E.: Efficient top-k similarity document search utilizing distributed file systems and cosine similarity. Cluster Computing **19**(1), 109–126 (2016). DOI 10.1007/s10586-015-0506-0. URL http://dx.doi.org/10.1007/s10586-015-0506-0
2. Allahbakhsh, M., Benatallah, B., Ignjatovic, A.: Quality control in crowdsourcing systems. IEEE INTERNET COMPUTING pp. 76–81 (2013)
3. Chen, J.J., Menezes, N.J., Bradley, A.D., North, T.: Opportunities for crowdsourcing research on amazon mechanical turk. Interfaces **5**(3) (2011)
4. Dai, P., Lin, C.H., Weld, D.S., et al.: Pomdp-based control of workflows for crowdsourcing. Artificial Intelligence **202**, 52–85 (2013)
5. Doan, A., Ramakrishnan, R., Halevy, A.Y.: Crowdsourcing systems on the world-wide web. Communications of the ACM **54**(4), 86–96 (2011)
6. Franklin, M.J., Kossmann, D., Kraska, T., Ramesh, S., Xin, R.: Crowddb:answering queries with crowdsourcing. In: SIGMOD, pp. 61–72 (2011)

7. Gomes, C., Schneider, D., Moraes, K., de Souza, J.: Crowdsourcing for music: Survey and taxonomy. In: Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on, pp. 832–839 (2012)
8. Howe, J.: The rise of crowdsourcing. Wired magazine **14**(6), 1–4 (2006)
9. Hofeld, T., Tran-Gia, P., Vucovic, M.: Crowdsourcing: From theory to practice and long-term perspectives (dagstuhl seminar 13361). Dagstuhl Reports **3**(9), 1–33 (2013)
10. Hu, J., Liu, Y.: Analysis of documents born digital. In: Handbook of Document Image Processing and Recognition, pp. 775–804. Springer London (2014). DOI 10.1007/978-0-85729-859-1_26
11. Huang, F., Li, J., Lu, J., Ling, T.W., Dong, Z.: Pandasearch: a fine-grained academic search engine for research documents. In: ICDE 2015 (2015)
12. Hung, N.Q.V., Tam, N.T., Tran, L.N., Aberer, K.: An evaluation of aggregation techniques in crowdsourcing. In: Web Information Systems Engineering–WISE 2013, pp. 1–15. Springer (2013)
13. Ipeirotis, P.G., Provost, F., Wang, J.: Quality management on amazon mechanical turk. In: Proceedings of the ACM SIGKDD Workshop on Human Computation, HCOMP '10, pp. 64–67. ACM, New York, NY, USA (2010). DOI 10.1145/1837885.1837906
14. Joglekar, M., Garcia-Molina, H., Parameswaran, A.: Evaluating the crowd with confidence. In: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 686–694 (2013)
15. Kamar, E., Hacker, S., Horvitz, E.: Combining human and machine intelligence in large-scale crowdsourcing. In: AAMAS, pp. 467–474 (2012)
16. Kittur, A., Nickerson, J.V., Bernstein, M., Gerber, E., Shaw, A., Zimmerman, J., Lease, M., Horton, J.: The future of crowd work. In: Proceedings of the 2013 conference on Computer supported cooperative work, pp. 1301–1318 (2013)
17. Klampfl, S., Granitzer, M., Jack, K., Kern, R.: Unsupervised document structure analysis of digital scientific articles. International Journal on Digital Libraries **14**(3), 83–99 (2014)
18. Kondreddi, S.K., Triantafillou, P., Weikum, G.: Combining information extraction and human computing for crowdsourced knowledge acquisition. In: ICDE, pp. 988–999 (2014)
19. Kulkarni, A.: The complexity of crowdsourcing: Theoretical problems in human computation. In: CHI Workshop on Crowdsourcing and Human Computation (2011)
20. Kuncheva, L.I., Whitaker, C.J., Shipp, C.A., Duin, R.P.W.: Limits on the majority vote accuracy in classifier fusion. Pattern Analysis & Applications **6**(1), 22–31 (2003)
21. Li, P., yang Yu, X., Liu, Y., ting Zhang, T.: Crowdsourcing fraud detection algorithm based on ebbinghaus forgetting curve. International Journal of Security & Its Applications **8**(1), 283 (2014)
22. Lofi, C., Maarry, K.E.: Design patterns for hybrid algorithmic-crowdsourcing workflows. In: CBI, pp. 1–8 (2014)
23. Luz, N., Silva, N., Novais, P.: Generating human-computer micro-task workflows from domain ontologies. In: Human-Computer Interaction. Theories, Methods, and Tools, pp. 98–109. Springer (2014)
24. Luz, N., Silva, N., Novais, P.: A survey of task-oriented crowdsourcing. Artificial Intelligence Review pp. 1–27 (2014). DOI 10.1007/s10462-014-9423-5
25. Mozafari, B., Sarkar, P., Franklin, M.J., Jordan, M.I., Madden, S.: Scaling up crowd-sourcing to very large datasets: A case for active learning. Proceedings of the VLDB Endowment (PVLDB) **8**(2), 125–136 (2014)
26. PANOS, I., LITTLE, G., MALONE, T.W.: Composing and analyzing crowdsourcing workflows. Collective Intelligence pp. 1–3 (2014)
27. Quinn, A.J., Bederson, B.B.: Human computation: a survey and taxonomy of a growing field. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1403–1412 (2011)
28. Rzeszotarski, J., Kittur, A.: Crowdscape: interactively visualizing user behavior and output. In: Proceedings of the 25th annual ACM symposium on User interface software and technology, pp. 55–62 (2012)
29. Sabou, M., Bontcheva, K., Derczynski, L., Scharl, A.: Corpus annotation through crowdsourcing: Towards best practice guidelines. In: Proc. LREC (2014)
30. Saxton, G.D., Oh, O., Kishore, R.: Rules of crowdsourcing: Models, issues, and systems of control. Information Systems Management **30**(1), 2–20 (2013)
31. Swaraj, K.P., Manjula, D.: A fast approach to identify trending articles in hot topics from xml based big bibliographic datasets. Cluster Computing **19**(2), 837–848 (2016). DOI 10.1007/s10586-016-0561-1. URL `http://dx.doi.org/10.1007/s10586-016-0561-1`
32. Wang, G., Wang, T., Zheng, H., Zhao, B.Y.: Man vs. machine: Practical adversarial detection of malicious crowdsourcing workers. In: 23rd USENIX Security Symposium, USENIX Association, CA, pp. 239–254 (2014)
33. 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)
34. Yin, X., Liu, W., Wang, Y., Yang, C., Lu, L.: What? how? where? a survey of crowdsourcing. In: Frontier and Future Development of Information Technology in Medicine and Education, *Lecture Notes in Electrical Engineering*, vol. 269, chap. 22, pp. 221–232. Springer Netherlands (2014). DOI 10.1007/978-94-007-7618-0_22
35. Zhao, Y., Zhu, Q.: Evaluation on crowdsourcing research: Current status and future direction. Information Systems Frontiers pp. 1–18 (2014)

Zhaoan Dong is a PhD student of the School of Information, Renmin University of China. He got his M.S. at East China Normal University in 2005. Then he joined Qufu Normal University and had worked there for eight years. His current research interests include knowledge graph completion and crowdsourcing.

Jiaheng Lu is an associate professor of the Department of Computer Science at the University of Helsinki, Finland. He got his Ph.D. degree at the National University of Singapore. He did two year Postdoc research at the University of California, Irvine. Then he joined the Renmin University of China and had worked there for seven years. His recent research interests include multi-model database management systems, semantic string processing and job optimization for big data platform. He has published more than 50 papers in database conferences and journals including SIGMOD, VLDB, ICDE, TODS, etc. He has served as a program member in SIGMOD, VLDB and ICDE, etc.

Tok Wang Ling is a professor of the Department of Computer Science, School of Computing at the National University of Singapore. He received his Ph.D. and M.Math., both in Computer Science, from University of Waterloo (Canada) and B.Sc.(1st class Hons) in Mathematics from Nanyang University (Singapore). His research interests include Data Modeling, Entity-Relationship Approach, Object-Oriented Data Model, Normalization Theory, Logic and Database, Integrity Constraint Checking, Semi-Structured Data Model, XML Twig Pattern Query Processing, ORA-semantics based XML and Relational Database Keyword Query Processing.

Ju Fan is an associate professor of the School of Information, Renmin University of China. He received his PhD in Computer Science from Tsinghua University in 2012. Then he did Postdoc research in the DBSystem Lab at the National University of Singapore from 2012 to 2015. His current research interests include crowdsourcing, data curation, and big data analytics. He has published more than 20 papers in top conferences and journals. He has served as a reviewer in database conferences including SIGMOD, VLDB and ICDE and program committee member in SIGMOD, ACM MM and WAIM.

Yueguo Chen is an associate professor of the School of Information, Renmin University of China. He received the BS and MS from Tsinghua University, Beijing, in 2001 and 2004. He earned his PhD in Computer Science from the National University of Singapore in 2009. His recent research interests include interactive analysis of big data, large-scale RDF knowledge base management.