# PANDA: A Platform for Academic Knowledge Discovery and Acquisition

Zhaoan Dong*, Jiaheng Lu†*, Tok Wang Ling‡
* DEKE, MOE and School of Information, Renmin University of China, Beijing, China
†Department of Computer Science, University of Helsinki, Finland
‡Department of Computer Science, School of Computing, National University of Singapore, Singapore
*Email: dongzhaoan@163.com; jiahenglu@gmail.com; lingtw@comp.nus.edu.sg*

***Abstract*—Scientific literatures contain some academic knowledge which is interesting or valuable but previously unknown. For instance, an algorithm A proposed in one article might have association with algorithm B in another article, while algorithm B is designed based on the definition of C in a third article. Thus we can deduce the relationship A-C based on A-B and B-C. There are also other kinds of academic knowledge such as association between two research communities, historical evolvement of a research topics, etc. But with the exponential growth of research articles that usually published in Portable Document Format (PDF), to discover and acquire potential knowledge poses many practical challenges. Existing algorithmic methods can hardly extend to handle diverse journals and layouts, nor scale up to process massive documents. As crowdsourcing has become a powerful paradigm for problem-solving especially for tasks that are difficult for computer to resolve solely, we state the problem of academic knowledge discovery and acquisition using an hybrid framework, integrating the accuracy of human workers and the speed of automatic algorithms. We briefly introduce a Platform for Academic kNowledge Discovery and Acquisition (PANDA), our current system implementation, as well as some preliminary achievements and promising future directions.**

## I. MOTIVATION AND CHALLENGES

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 like Google Scholar [1] through a user-friendly search interface. And they always 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 past 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 one by one. It is very time-consuming and costly especially when some papers are found useless and dropped at last.

Fortunately, tremendous interests have been given to extraction and management of research data. For example, *Digital Curation* (DC) [2] indicates both activities required to maintain research data long-term and the process of extraction of important information from scientific literature. Another example is *Deep Indexing*(DI) [3] which indexes the research data within articles that are usually invisible to the traditional bibliographic searches. *Deep Indexing* is now available in ProQuest [4], CiteSeer$X$ [5] and ScienceDirect [6], etc.

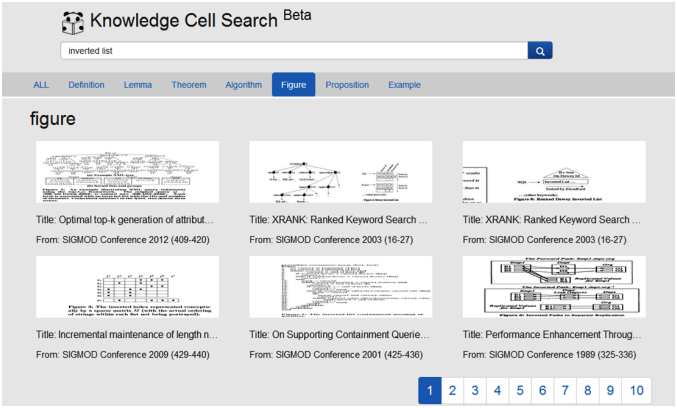



Fig. 1: Knowledge Cell Search Results of Pandasearch

Recently, we have proposed a novel academic search engine named PandaSearch [7], [8]. As is shown in Fig. 1, when users submit a keyword like "*inverted list*", the system returns a list of meaningful information objects like Definitions, Figures, Lemmas, Theorems and Algorithms that are most relevant to the keyword. All of these meaningful objects will be defined as "**Knowledge Cells**" in Section 3. Another important definition that will be given in Section 3 is "**Academic Knowledge Graph**", which is crucial for further academic knowledge discovery and exploration based on the relationships of the Knowledge Cells. The relationships are usually implied or hidden in the sentences of the article. For example, if an author writes "*We continue the example of Figure XXX to illustrate the algorithm of ...*" in a paper, he usually indicate the relationship between a Figure and an Algorithm in the same article. And another sentence like "*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.

Obviously, the most important prerequisite for building the Academic Knowledge Graph is to correctly identify and extract Knowledge Cells including the names, contents and contexts, as well as various relationships among them. However, we have to face several challenges as follows to achieve the above objectives.

The first challenge is to identify and extract each Knowledge Cell correctly. Although PDF has become the de facto standard of science literature, a scientific document is more

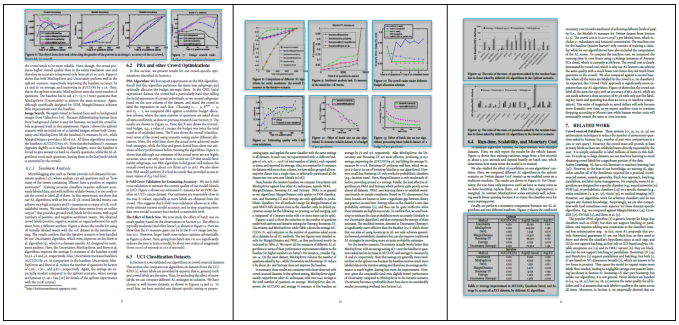


Fig. 2: An Example of Different Layouts

complex than it seems. Human can easily deduce structure and semantics of the different characters and pictures on a page, but it is hard for computer algorithms. The main reason is PDFs intrinsically do not contain or store enough structural information, and they only provide the rendering information of individual text fragments for final presentation.

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 [9]. 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 tremendous amount of science literature rapidly growing.

For example, in Fig. 2, we selected page 9-11 from [10]. There are at least three different layouts of 11 logical objects including one Table and ten Figures. Therefore, this poses a cumbersome task to current rule-based or learning-based extracting algorithms.

The second challenge is to extract the contents, key phrases and contexts of Knowledge Cells. Sometimes, important information about Knowledge Cells is implied in the captions of Figures, specifications of Algorithms, and sometimes the content of a Knowledge Cell is only an illustration, which is hard for a computer to understand. This poses a great challenge for algorithms to automatically extracting solely.

The third challenge is to build an Academic Knowledge Graph to represent the Knowledge Cells and their relationships. In practice, some relationships may be implied in the contents and contexts of Knowledge Cells. But 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 when performed fully automatically.

Crowdsourcing is an promising on-line problem solving paradigm tapping the intellect of the crowd. Crowdsourcing platforms such as Mechanical Turk have been widely applied to solving various tasks such as data collection, image labeling, recognition and categorization, translation [11], etc. Additionally, Human Computation has gained renewed interest in solving complex tasks that cannot be easily addressed by automatic algorithms [12]. 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 performance. On the other hand, if a great number of PDFs are crowd-sourced, 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.

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

## II. Related Work

### A. Automatic Information Extraction

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 [13]. In these methods, the PDF pages are firstly converted into images as pre-processing 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 [9]. Statistical methods and Artificial Intelligence techniques, including Probabilistic Modeling, Naïve Bayes Classifier and Conditional Random Field, Support Vector Machines are widely used [14]. Optical Character Recognition and Natural Language Processing techniques are also necessary for textual information extraction.

Another way is 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 [13]. 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 [15] can be exploited.

### B. Task-Oriented Crowdsourcing

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 [11], [16]–[22]. In computer science, crowdsourcing is highly connected

to human computation [17], [23], [24], which replacing machines with humans in certain computational steps where humans usually perform better. Just as stated in [17] 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.

*1) Crowdsourcing task and workflow:* A central challenge of crowdsourcing is how to design tasks within the expected monetary costs and results quality in mind. According to [11], crowdsourcing tasks can be categorized into two types: *micro-tasks* and *complex-tasks*. While *micro-tasks* are atomic operations, *complex-tasks* are organized sets (e.g. *workflows*) of micro-tasks with a specific purpose. When solving complex tasks, different methods (e.g.*crash* and *rerun*,*map reduce*,*divide* and *conquer*) can be utilized to manage workflows of tasks [11]. Some of the predefined templates or design patterns for task design, workflow design, and reviewing methodologies have been provided by Mechanical Turk [25]. In the most recent, Sabou et al. [26] proposed a set of best practice guidelines for crowdsourcing task design. Luz et al. [27] 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. [12]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) 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) [28], 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) [29], a series of iterations will be performed. Each iteration contains two steps [30]: (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 [30] 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*.

*3) Quality control:* Nowadays, researchers have developed some mechanisms to detect malicious behavior and fraud. For example, Rzeszotarski et al. [31] presented *CrowdScape*, a system that supports the human evaluation of complex tasks through interactive visualization and mixed initiative machine learning. Joglekar et al. [32]devised techniques to generate confidence intervals for worker error estimates. Allahbakhsh et al. [33] proposed a general framework for characterizing two main dimensions of quality control: worker profiles and task design. Dai et al. [34] and Panos et al. [35] separately devoted themselves to analyzing and optimizing existing workflows to improve both the quality and the cost of crowdsourcing.

In most recent, Li et al. [36] put forward a crowdsourcing fraud detection method to find out the spammer according to the psychological difference. Wang et al. [37] developed a machine learning model against practical adversarial attacks in the context of detecting malicious crowdsourcing activity.

### C. Human Computation for Information Extraction

With the advent of human computation and crowdsourcing, some entities have devoted themselves to human-computer workflows for information processing.

For example, Kamar [38] 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. [12] extensively investigated hybrid crowdsourcing human computation workflows and abstracted five generic design patterns: *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 [39] 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. [10] proposed two Active Learning algorithms for labeling tasks in crowd-sourced databases, MinExpError and Uncertainty, to decide which items should be sent 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.

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

## III. PROBLEM STATEMENT

We firstly give 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 which paper is this knowledge cell from), 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 keyphrases (i.e. the reference contexts of a Knowledge Cell which are usually some sentences or paragraphs). Specially, 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 the Knowledge Cells and arrows with different labels to represent various relationships. With the aid of Academic Knowledge Graph, our academic search engine, i.e. PandaSearch, can provide a fine-grained search as is shown in Fig. 1 in addition to traditional academic searches. In the future, on the one hand, we intend to add "**Advanced Search**" to PandaSearch through form-based UIs for common users. On the other hand, we can provide SQL-Like APIs for external systems as demonstrated in the following examples. [1].

**Example 1**: To find the Figures that contain "inverted list" in their names. At the same time, we also want to know which papers are they from.

```
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, we should find **Figures** from Knowledge Cells containing "*inverted list*" in their names. 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**: Search algorithms which are variants or have been compared with an **Algorithm** whose name is related to "*hash join*" algorithm. We hope the two Algorithms mentioned above are from different papers.

[1]We use two non-standard SQL statements to symbolically 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.

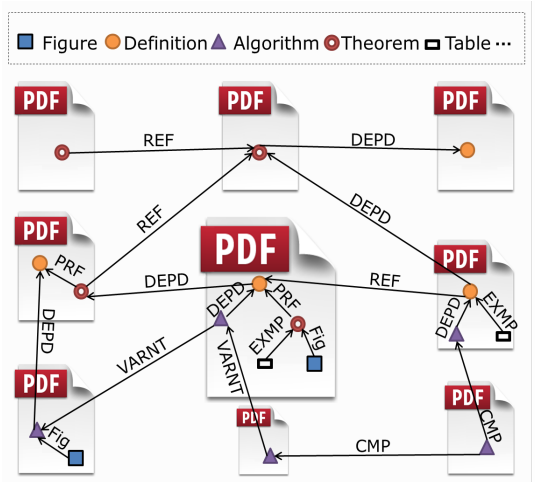



Fig. 3: A Fragment of an Academic Knowledge Graph

```
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, if any, where **CMP** can represent *comparison* relationship between $k_1$ and $k_2$ and **VARNT** means $k_1$ is a *variant* or *extension* of $k_2$.

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. See Fig. 3) can be manually defined by human workers with hints/guidances or automatically by heuristic rules in the future extraction process. As described above, we can state the problem as follows.

**Problem statement.** In this research, the problem of academic knowledge discovery and acquisition can be modeled as a crowd-sourced database problem [40], where scholarly papers, Knowledge Cells and the relationship between Knowledge Cells 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 transparently combine the automatic algorithms and crowdsourced tasks.

## IV. An Overview of Academic Knowledge Acquisition Framework

We propose a generic framework for academic knowledge discovery and acquisition from PDFs as a multi-stage process. We briefly describe the framework in this section.

**(1) Preprocessing stage.** In this stage, PDF documents are first preprocessed for later stages. Firstly, for example, metadata information of a paper such as title, author, publication, and pages could be harvested from DBLP, Google Scholar, etc. in advance. Additionally, in order to perform text analysis

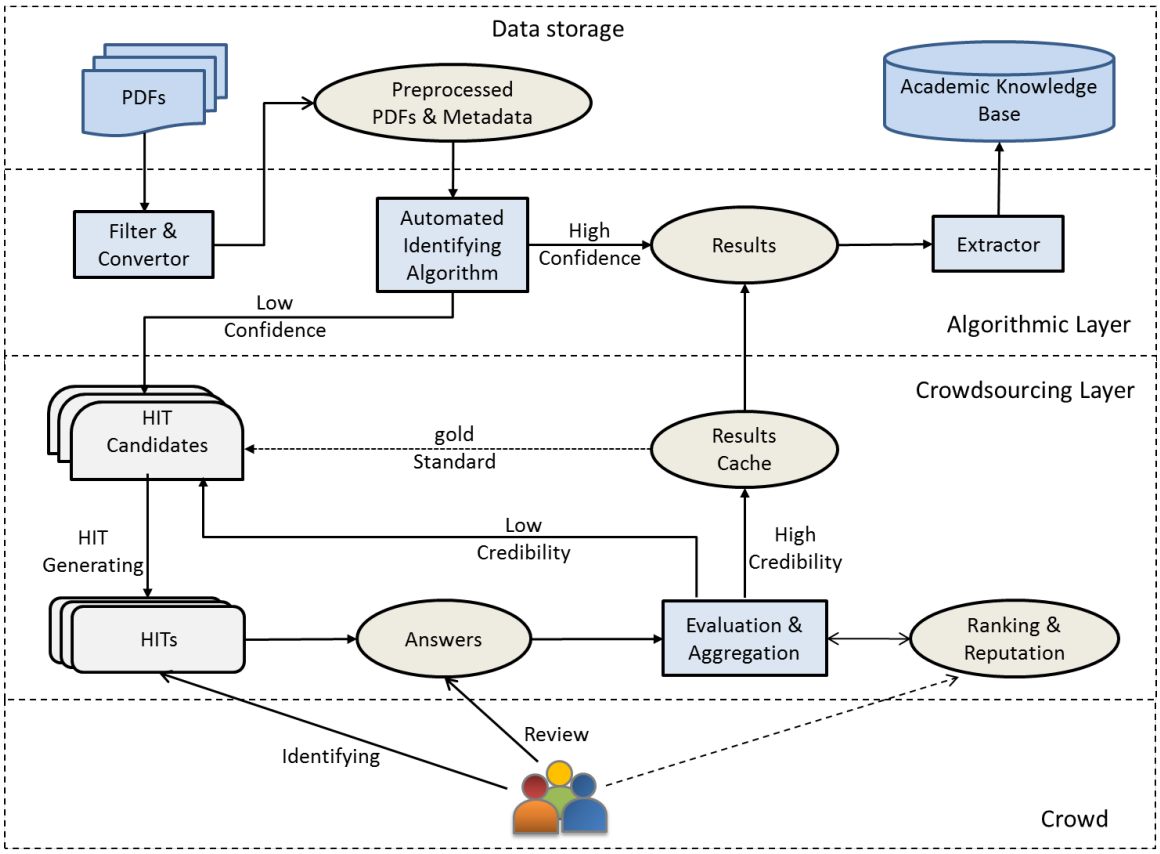



Fig. 4: The Architecture of the Prototype

to extract the topics and contexts of each Knowledge Cell, the PDF documents should be converted to a standard textual format. Moreover, it is necessary for PDF documents to be split into pages for automatic extraction and Human Intelligent Tasks generation. Some PDF pages that obviously do not contain the targets should be filtered by rule-based filters for each kind of Knowledge Cells.

**(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 estimate on how accurate and reliable a 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). In the next steps, Human Intelligence Tasks (HITs) for extracting certain Knowledge Cell will be designed and generated based on the set of Human Intelligence Task Candidates. 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.** Based on the hybrid human-computer work-flows, we try to build a task-oriented crowdsourcing system. Human workers would be recruited for 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 and cost model.** 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 participate 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. Based on the cost model, we could study how to archive a higher quality with a fixed budget, or oppositely, how to reduce the cost with quality constraints.

## V. System Implementation

In this section, we will introduce the system implementation with some preliminary experimental results. Our system, named PANDA (abbr. of the Platform for Academic kNowledge Discovery and Acquisition), has served as a data provider for PandaSearch [7]. The system architecture of PANDA, as is shown in Fig. 4, can be divided into 4 layers.

TABLE I: Statistics of Current Data Stores

| Data Type | Number |
|---|---|
| Papers | 2975828 |
| Figures | 15492 |
| Definitions | 1939 |
| Lemmas | 757 |
| Theorems | 726 |
| Algorithms | 671 |
| Propositions | 52 |
| Examples | 1,038 |

**Algorithm 1** AutoExtractingFigures.

**Input:** *A PDF document, D.*
**Output:** *The locations of rectangles containing Figures, R.*

```
PDFpages←spliter(D);
TextFile←convertor(D);
FilteredPages←rule-based-filter(PDFpages);
while FilteredPages.hasMore() do
    CurPage←FilteredPages.nextPage();
    Locations←Rule-based-Locating(CurPage,TextFile);
    while Locations.hasMore() do
        curPostion←Locations.nextPosition();
        (UpLeftX,UpLeftY,LowRightX,LowRightY)
        ←Boundary_Detector(CurPage,curPosition);
        R←R ∪ (UpLeftX,UpLeftY,LowRightX,LowRightY);
    end while
    return R;
end while
```

### A. Data Storage

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

### B. Algorithmic Layer

Currently, we have built an algorithm using rule-based and machine learning methods to automatically extract Figures. As is shown in Algorithm 1, the first step is to split the PDF document into pages. And then a set of rules are exploited to filter the pages that obviously do not contain Figures. For example, the rules indicating one PDF page has no Figures may be: (i) It is a "cover page" or a "title page" of the PDF document, (ii) It begins with a new bibliography item which means the beginning of bibliography section. (iii) The number of identified lines is MAXIMUM which means there is no space for any Figures. Oppositely, some rules indicating at least one "Figure" in a PDF page: (i) There is a word "Figure" or "Fig", (ii) The word is followed by a number, etc.

We have designed a boundary detector to identify the positions and the boundary rectangle of the 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 [15]. We also take advantage of the open source libSVM [41] 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.

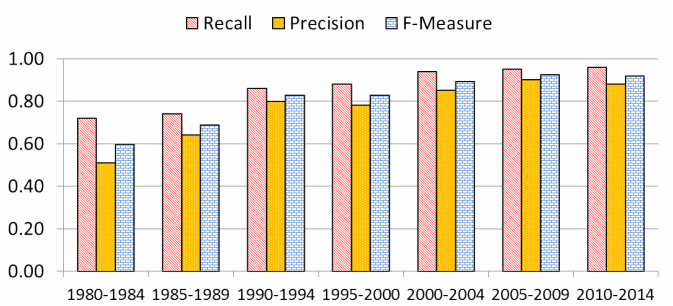



Fig. 5: Performance of Automatic Extracting Figures.

The up-left and low-right corners of a bounding rectangle are computed by a machine learning algorithm, and then sent to a image-cutter ( i.e. the Extractor in Fig. 4) for segmenting.

We perform an initial experiment for extracting Figures, based on 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:

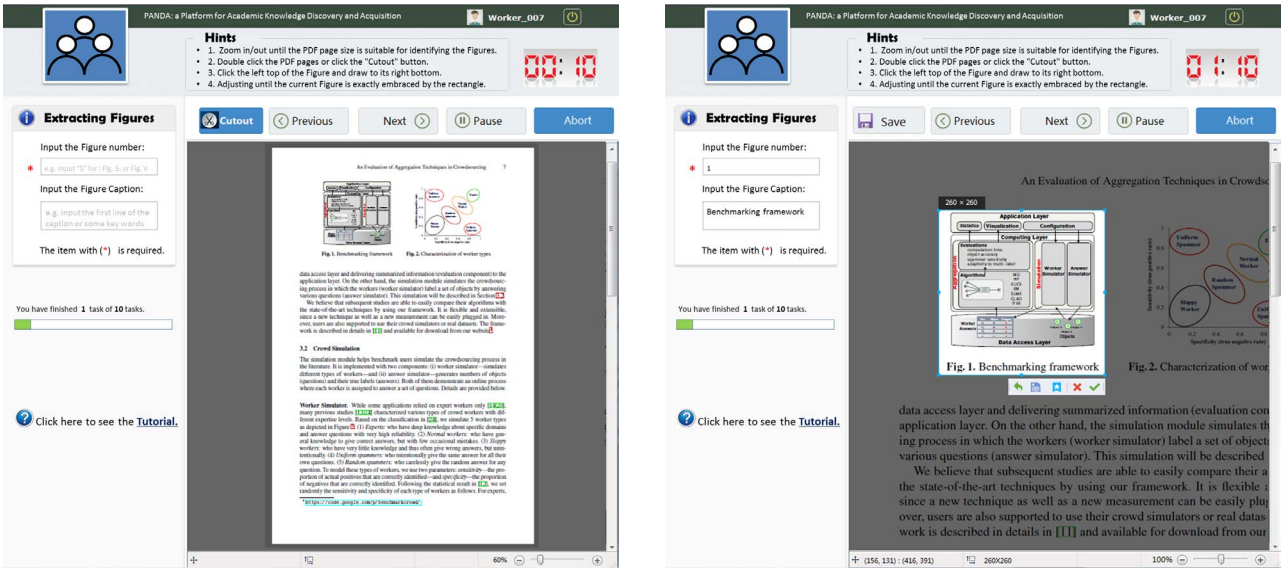
$$Recall = \frac{\#correctly\ identified\ Figures}{\#Figures\ in\ the\ paper}$$

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

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

Fig. 5 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 I, 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 [7] for average, are still under development and need to be further optimized. Hence we do not describe them here due to the space limitation.




(a) Before Identifying the Boundaries

(b) After Identifying the Boundaries

Fig. 6: An Example of Web-based Interfaces for Extracting Figures.

### C. Crowdsourcing Layer

To handle pages with low confidence, crowdsourcing tasks would be designed and generated by the HIT generator. After being accomplished 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. Recently, several basic crowdsourcing workflows have been developed. For example, crowdsourcing tasks for extracting Figures can be divided in to two categories: identifying and review.

(1) Identifying: Human workers segment the graphical regions of Knowledge Cells and input some descriptions according to the hints by browsing PDF pages one by one. As shown in Fig. 6a, the worker can click the “Cutout ” button to refine the bounding rectangle of one Figure by dragging the mouse. The worker can also be asked to input the number and the Caption of the Figure through the left input form when the PDF page has low quality, i.e. too difficult for algorithms to extract those information. At last, the results can be saved by clicking the “Save” button. All the operations must be finished within a time limit, 5 minutes for example. In order to keep the workers being active, tasks assigned to each worker should be not too much. We allocate 10 tasks to each worker in the example of Extracting Figure.

(2) Review: The goal of review tasks is to evaluate the answers contributed by other workers. Those human workers who have accomplished the identifying tasks with high performance have opportunities to apply the review tasks. For some binary review tasks, we currently simplify the method of Majority Vote to three reviewers at most for each task. For example, a reviewer may be asked whether the graphical component of a knowledge cell have been well segmented. 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 methods 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. For more complex tasks, we are now trying out best to design corresponding review methods including breaking them into some simpler review tasks. Moreover, in order to save the network bandwidth, we transfer the textual information and the coordinateness to the Extractor instead of raw images directly. The Extractor in Fig. 4 will cutout and save the images according to the coordinateness.

### D. Crowd

In this part, the challenge is how to motivate and retain an appropriate group of human workers. As for current system, we recruit 30 student volunteers to try out the system for an initial training set. We are now developing a user management module which can provide functions for each human worker to register and view the ranking and reputation.

## VI. Conclusion and Future Plan

We describe a hybrid framework for academic knowledge discovery and acquisition integrating the accuracy of human workers and the speed of automatic algorithms. On the one hand, we make use of rule-based and machine learning methods as well as open source software for identifying and

extracting Knowledge Cells. On the other hand, we develop a web-based crowdsourcing module for Figure extraction.

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) Strategies** which can be used to switch tasks between algorithms and crowdsourcing module; **(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 fixed budget, or reducing the whole cost in terms of time and money with quality constraints. Finally, we will extend the framework to identify and extract the contents, key phrases and contexts of Knowledge Cells, as well as the relationships among them to construct the Academic Knowledge Graph. Together with PandaSearch, 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.

## ACKNOWLEDGMENT

This research work is partially supported by NSF China (No.61472427) and RUC Research Funds (No. 11XNJ003).

## REFERENCES

[1] GoogleScholar, "Googlescholar," http://scholar.google.com/.
[2] Wikipedia, "Digital curation," http://http://en.wikipedia.org/wiki/Digital_curation.
[3] ProQuest, "Illustrata deep indexing," http://proquest.libguides.com/deepindexing.
[4] ——, "Proquest," http://search.proquest.com.
[5] CiteSeer, "Citeseer," http://citeseer.ist.psu.edu/.
[6] ScienceDirect, "Sciencedirect," http://www.sciencedirect.com/.
[7] F. Huang, J. Li, J. Lu, T. W. Ling, and Z. Dong, "Pandasearch: a fine-graned academic search engine for research documents," in *ICDE 2015*, 2015.
[8] PandaSearch, "Pandasearch," http://pandasearch.ruc.edu.cn/.
[9] S. Klampfl, M. Granitzer, K. Jack, and R. Kern, "Unsupervised document structure analysis of digital scientific articles," *International Journal on Digital Libraries*, vol. 14, no. 3, pp. 83–99, 2014.
[10] B. Mozafari, P. Sarkar, M. J. Franklin, M. I. Jordan, and S. Madden, "Scaling up crowd-sourcing to very large datasets: A case for active learning," *PVLDB*, vol. 8, no. 2, pp. 125–136, 2014.
[11] N. Luz, N. Silva, and P. Novais, "A survey of task-oriented crowdsourcing," *Artificial Intelligence Review*, pp. 1–27, 2014.
[12] C. Lofi and K. E. Maarry, "Design patterns for hybrid algorithmic-crowdsourcing workflows," in *CBI*, 2014, pp. 1–8.
[13] J. Hu and Y. Liu, "Analysis of documents born digital," in *Handbook of Document Image Processing and Recognition*. Springer London, 2014, pp. 775–804.
[14] J. Wu, K. Williams, H. Chen, M. Khabsa, C. Caragea, A. Ororbia, D. Jordan, and C. L. Giles, "Citeseerx: AI in a digital library search engine," in *AAAI*, 2014, pp. 2930–2937.
[15] PDFBox, "Pdfbox," http://pdfbox.apache.org/.
[16] A. Doan, R. Ramakrishnan, and A. Y. Halevy, "Crowdsourcing systems on the world-wide web," *Communications of the ACM*, vol. 54, no. 4, pp. 86–96, 2011.
[17] A. J. Quinn and B. B. Bederson, "Human computation: a survey and taxonomy of a growing field," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2011, pp. 1403–1412.
[18] G. D. Saxton, O. Oh, and R. Kishore, "Rules of crowdsourcing: Models, issues, and systems of control," *Information Systems Management*, vol. 30, no. 1, pp. 2–20, 2013.
[19] T. Hofeld, P. Tran-Gia, and M. Vucovic, "Crowdsourcing: From theory to practice and long-term perspectives (dagstuhl seminar 13361)," *Dagstuhl Reports*, vol. 3, no. 9, pp. 1–33, 2013.
[20] A. Kittur, J. V. Nickerson, M. Bernstein, E. Gerber, A. Shaw, J. Zimmerman, M. Lease, and J. Horton, "The future of crowd work," in *Proceedings of the 2013 conference on Computer supported cooperative work*, 2013, pp. 1301–1318.
[21] Y. Zhao and Q. Zhu, "Evaluation on crowdsourcing research: Current status and future direction," *Information Systems Frontiers*, pp. 1–18, 2014.
[22] X. Yin, W. Liu, Y. Wang, C. Yang, and L. Lu, "What? how? where? a survey of crowdsourcing," in *Frontier and Future Development of Information Technology in Medicine and Education*, ser. Lecture Notes in Electrical Engineering. Springer Netherlands, December 2014, vol. 269, ch. 22, pp. 221–232.
[23] A. Kulkarni, "The complexity of crowdsourcing: Theoretical problems in human computation," in *CHI Workshop on Crowdsourcing and Human Computation*, 2011.
[24] C. Gomes, D. Schneider, K. Moraes, and J. de Souza, "Crowdsourcing for music: Survey and taxonomy," in *Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on*, 2012, pp. 832–839.
[25] J. J. Chen, N. J. Menezes, A. D. Bradley, and T. North, "Opportunities for crowdsourcing research on amazon mechanical turk," *Interfaces*, vol. 5, no. 3, 2011.
[26] M. Sabou, K. Bontcheva, L. Derczynski, and A. Scharl, "Corpus annotation through crowdsourcing: Towards best practice guidelines," in *Proc. LREC*, 2014.
[27] N. Luz, N. Silva, and P. Novais, "Generating human-computer micro-task workflows from domain ontologies," in *Human-Computer Interaction. Theories, Methods, and Tools*. Springer, 2014, pp. 98–109.
[28] L. I. Kuncheva, C. J. Whitaker, C. A. Shipp, and R. P. W. Duin, "Limits on the majority vote accuracy in classifier fusion," *Pattern Analysis & Applications*, vol. 6, no. 1, pp. 22–31, 2003.
[29] P. G. Ipeirotis, F. Provost, and J. Wang, "Quality management on amazon mechanical turk," in *Proceedings of the ACM SIGKDD Workshop on Human Computation*, ser. HCOMP '10. New York, NY, USA: ACM, 2010, pp. 64–67.
[30] N. Q. V. Hung, N. T. Tam, L. N. Tran, and K. Aberer, "An evaluation of aggregation techniques in crowdsourcing," in *Web Information Systems Engineering–WISE 2013*. Springer, 2013, pp. 1–15.
[31] J. Rzeszotarski and A. Kittur, "Crowdscape: interactively visualizing user behavior and output," in *Proceedings of the 25th annual ACM symposium on User interface software and technology*, 2012, pp. 55–62.
[32] M. Joglekar, H. Garcia-Molina, and A. Parameswaran, "Evaluating the crowd with confidence," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2013, pp. 686–694.
[33] M. Allahbakhsh, B. Benatallah, and A. Ignjatovic, "Quality control in crowdsourcing systems," *IEEE INTERNET COMPUTING*, pp. 76–81, 2013.
[34] P. Dai, C. H. Lin, D. S. Weld *et al.*, "Pomdp-based control of workflows for crowdsourcing," *Artificial Intelligence*, vol. 202, pp. 52–85, 2013.
[35] I. PANOS, G. LITTLE, and T. W. MALONE, "Composing and analyzing crowdsourcing workflows," *Collective Intelligence*, pp. 1–3, 2014.
[36] P. Li, X. yang Yu, Y. Liu, and T. ting Zhang, "Crowdsourcing fraud detection algorithm based on ebbinghaus forgetting curve," *International Journal of Security & Its Applications*, vol. 8, no. 1, p. 283, January 2014.
[37] G. Wang, T. Wang, H. Zheng, and B. Y. Zhao, "Man vs. machine: Practical adversarial detection of malicious crowdsourcing workers," in *23rd USENIX Security Symposium, USENIX Association, CA*, 2014, pp. 239–254.
[38] E. Kamar, S. Hacker, and E. Horvitz, "Combining human and machine intelligence in large-scale crowdsourcing," in *AAMAS*, 2012, pp. 467–474.
[39] S. K. Kondreddi, P. Triantafillou, and G. Weikum, "Combining information extraction and human computing for crowdsourced knowledge acquisition," in *ICDE*, 2014, pp. 988–999.
[40] M. J. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin, "Crowddb:answering queries with crowdsourcing," in *SIGMOD*, 2011, pp. 61–72.
[41] C.-J. Lin, "Chih-jen lin's home page," http://www.csie.ntu.edu.tw/~cjlin/.