ActiveCite: An Interactive System for Automatic Citation Suggestion

by

Zhou Shaoping

A thesis submitted
for the degree of Master of Science
Department of Computer Science
School of Computing
National University of Singapore
2010

Abstract

Citations are very important in academic writing as they support the ideas presented in a work. Many authors use citation software to insert citations while they are writing.

To be able to insert citations using current software, authors must specify the references they wish to cite or search online to find appropriate sources. The process is often tedious and disrupts the writing flow.

The goal of our software prototype, ActiveCite, is to minimize the disruption caused by inserting citations so that authors can concentrate on writing. It uses the existing text in the document to provide a framework for searching and suggesting citations and integrating them into the work.

ActiveCite's interface features breadcrumbs and previews that allow users to easily switch back and forth between citation and writing. ActiveCite also includes a shorthand notation for passing contextual information to the back-end system. It uses partial information from the document for known-item citations and can suggest citations using subject search.

The results of the user study we conducted confirms ActiveCite's usability and its potential as a helpful and intuitive tool to support academic writing.

Acknowledgments

First of all, I would like to thank my supervisor, Dr. Zhao Shengdong, for giving me the inspiration for this thesis and providing guidance in writing it.

I would also like to thank Yang Xin, who collaborated with me on this project and contributed generously to it. I am grateful for his comments on how to improve ActiveCite.

Finally, I would like to give special thanks to my parents, whose love I can never repay enough.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Dating back to the use of Shepard's Citations in the legal community in 1873, citation indexing has been used to help authors decide on what references to include in their work [33]. References are used to identify previous research whose theory, approaches, results, etc. impact an author's work.

A citation can be loosely defined as a reference to a published or an unpublished source. More precisely, it is an abbreviated alphanumeric expression embedded in the body of an intellectual work. It corresponds to an entry in the bibliographic references section and acknowledges the relevance of other work to the current one. The combination of the in-body citation and the bibliographic entry constitutes a citation (whereas bibliographic entries by themselves are not) [3]. Authors of academic writing add citations to avoid plagiarism as well as to provide further explanation for sections of their own work [16].

Many scientists and other academic researchers spend a tremendous amount of time searching for related literature. Since the number of publications increases at a yearly rate of 3.7% [18], incorporating sufficient and appropriate number of references becomes increasingly challenging, and can take up more time and effort from researchers. Hence, researchers often rely

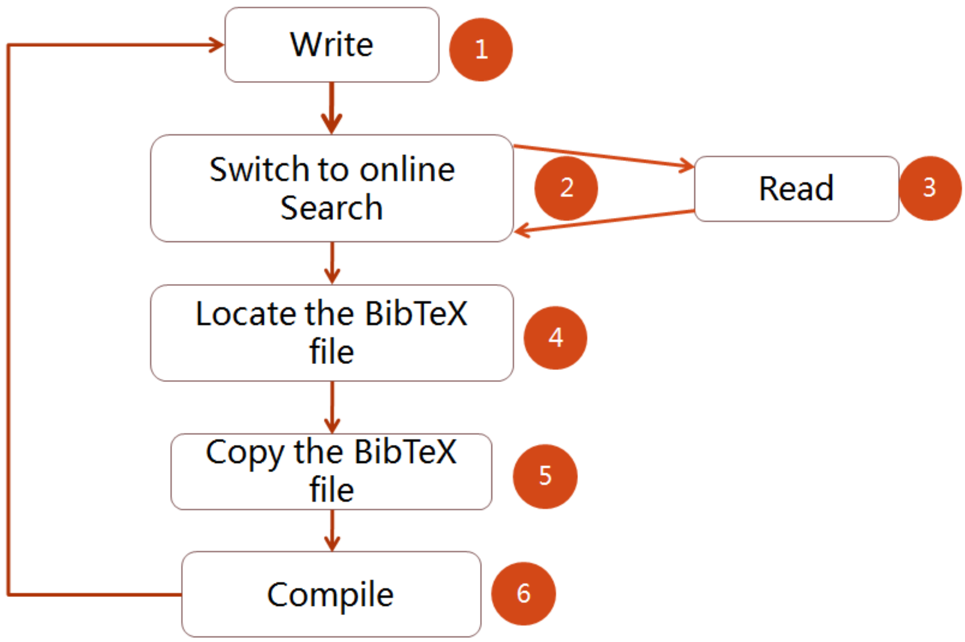


Figure 1.1: Typical workflow using LaTeX

on software citation management tools to organize relevant citations. The common software citation management tools include the BibTex file in LaTeX [15], EndNote, CiteULike [8], RefWorks, etc. These applications play a very important role in the writing process. However, most citations management tools today requires explicit and tedious management by the writers, and the citation management and insertion process often disrupt the writing process [17]. There is a need for better citation tools that are more integrated in the writing process and reduce the effort of management from writers.

Current software requires an author to specify the particular reference he wants to cite or to manually search online to find appropriate sources. LaTeX is a popular tool that supports this kind of citation management process. The typical workflow of LaTeX is shown in Figure 1.1.

Users noted that one of its limitations is that BibTex records of references that are not in the local bibliographic information database have to be searched online then copied there. This involves the actions 2, 3, 4, and 5 in Figure 1.1. Several iterations of actions 2 and 3 usually happen, causing a lot of disruptions when switching between searching and writing.

Based on information we gathered from the pilot interview, users' knowledge of citations can be roughly categorized into the following three categories: a known citation source, a roughly known citation source, and an unknown citation source.

The process of inserting a citation varies among users. Known citations are usually saved in personal archives, which can exist locally (e.g., a personal hard drive) or remotely (e.g., an online repository. They can be in the form of database records (e.g., BibTex files, EndNote) or files and can be easily inserted in the document. Roughly known citations and unknown citations, which often exist remotely, take more effort to access.

Most people are good at remembering something in a general sense rather than in detail(e.g., [7, 27]). This thesis aims to use general information authors know about their references to help them manage citation as they write. If an author saves the bibliographic details of all the references he has ever read, his local database will be bloated with references that are irrelevant to his current research. If he were to use such a database to cite a reference for a certain passage, he could get lost in the task of finding just one specific information and miss other relevant sources that he could also include. If he does not save them, he would have to go online to search manually using the partial information he has. The research question is thus, "how can this dilemma be solved?"

In this thesis, we present ActiveCite, an interactive system that allows users to make citation management easy and efficient. The interactions are designed with lesser disruptions to the writing process compared to traditional approaches used by other writing tools.

1.1 Contributions

This research introduces original techniques in the field of human computer interaction. Its three major contributions are: tight integration of writing and citation search, interaction techniques for citations, and automatic search term determination.

Tight Integration of Searching and Writing

Although there is previous research [4] on the integration of searching and writing, this thesis explores the subject further. ActiveCite allows users to postpone and resume the citation and writing process conveniently by tightly integrating citation search and writing. This is the most important contribution of our research.

Interaction Techniques for Citations

We proposed two interaction techniques, global suggestion and local suggestion, to allow citations to be inserted in the document easily and intuitively. Through these, a citation within the global suggestion window can be dragged and dropped into the document while a citation within the local suggestion window can be selected or deselected. These dramatically reduce the effort it takes to insert citations.

Automatic Search Term Determination

In the existing tools, an author has to input at least one search term in order to find relevant references. In ActiveCite, we introduce a new technique that automatically determines search terms based on the content before a particular citation marker. Apart from that, its global suggestion function also adopts the method of generating searching terms based on the changing content of the document. This has been done in a previous study [32].

1.2 Organization of the Thesis

Chapter 1 begins with the rationale behind the development of the software prototype and discusses its improvements on current citation software.

Chapter 2 gives an overview of the writing process, the techniques various citation software use to recommend references, and the existing solutions for these techniques' limitations.

Chapter 3 explains how we conducted our pilot interview and paper prototype evaluation. It also discusses the results of our preliminary work.

Chapter 4 takes us through a user's experience writing an academic paper using ActiveCite. The prototype's main features are also described in this chapter.

Chapter 5 details ActiveCite's specifications and other technical information.

Chapter 6 discusses how user evaluation for ActiveCite was done, its results and analysis.

Finally, Chapter 7 summarizes our work, discusses the limitations of our prototype and explores the directions we can take in the future.

Chapter 2

Related Work

This chapter reviews existing research on the problems encountered in citation management. The discussion begins with studies of the writing process, followed by the three classical methods for recommending references. Practical solutions for paper recommender systems also touch on how the interface and the techniques in citation recommendation have evolved.

2.1 Studies of the Writing Process

Academic writing is difficult because apart from the actual writing, it involves organizing research materials and gathering bibliographic information. The advancements in information search in digital libraries reduces the difficulty of preparing references. Current computer platforms also allow authors to integrate citation search and actual writing because they are done in different windows of the same computer.

However, the more research an author does, the harder it is to begin the actual writing [10]. It is a challenge faced by experts and novices alike [32]. The vast quantities of information in digital libraries turns the actions of searching articles and reading them into displacement

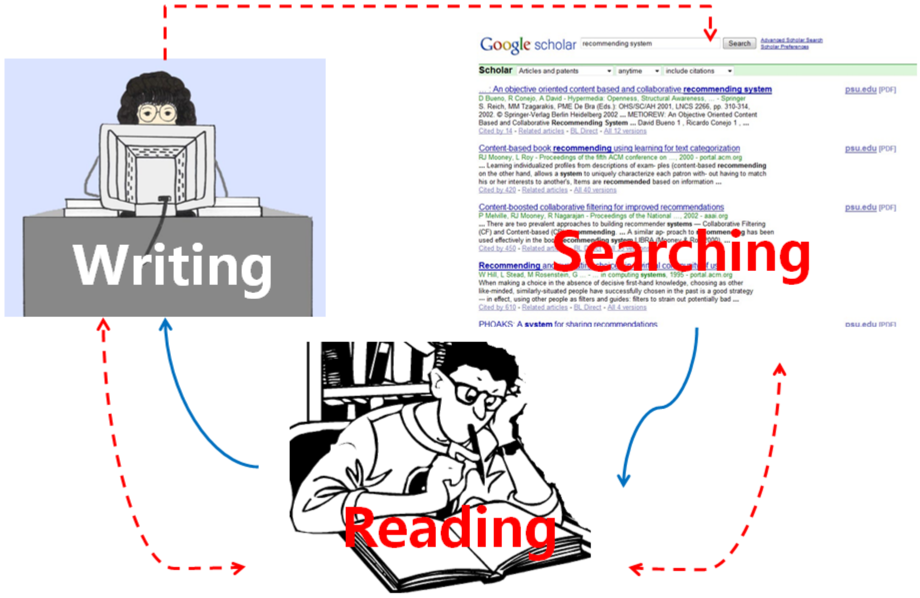


Figure 2.1: Dashed lines show the issue-driven approach while solid lines show the content-driven approach

activities. Experienced and perfectionist writers often postpone writing the paper when they get so much information.

Authors use two typical approaches in writing: issue-driven (writing down preliminary thoughts, looking for supportive sources, and reading) and content-driven (exhaustively searching for information, reading, and only then writing) [23]. As illustrated in Figure 2.1, the dashed line shows the issue-driven approach, which can be described as ”write while you search,” and the solid line shows the content-driven approach. More experienced writers prefer the issue-driven over the content-driven approach [23].

Some of the research on academic writing processes claim that tighter integration of writing and searching citations is one way of improving the quality of the final document. Writers often practice this, and evidence in Fister’s study [11] shows that even some successful students closely integrate searching, reading and writing. Thus, this thesis focuses on the tighter integration of these activities in order to minimize distraction from the actual writing.

2.2 Three Classical Methods for Recommending a Paper

As information retrieval/data mining (IR/DM) techniques continue to evolve, more methods for getting paper recommendations become available. Although there are no existing research paper recommender systems, one could be developed based on published and partly implemented concepts [13]. The process of recommending research papers generally involves identifying those that are similar to the one being written or are related to the keywords entered in a search.

Following is a brief discussion of the three classical recommendation techniques and their advantages and disadvantages.

2.2.1 Content-Based Technique

Recommendation systems based on content analysis are very popular in current academic search usage. The strength of popular academic search engines such as Google Scholar lies in classic text mining and in finding documents containing specific search terms or keywords.

However, researchers who search for articles using this approach encounter numerous problems because they have to deal with unclear nomenclatures, synonyms or context depending on the meaning of words [13]. Systems that use this technique often cannot recommend relevant references if different criteria are entered or when researchers are not sure about what keywords to search. This often delivers unsatisfactory results.

2.2.2 Collaborative-Based Technique

The collaborative-based technique involves recommending items based on ones liked by other users who have expressed similar preferences and that are not yet rated by the target user [24].

This has been used successfully in scenarios such as electronic commerce and information access. However, the use of this technique in research paper recommendation is criticized for various reasons [13]. Some authors say that this approach would be ineffective in cases where the number of items is more than the number of users [1] since the items that do not have user ratings cannot be recommended. Others claim that authors would be unwilling to spend time rating research papers [31].

Ratings could be directly obtained by considering citations as ratings [31] or implicitly generated by monitoring readers' actions (e.g., bookmarking or downloading a paper) [22, 25]. To get implicitly generated ratings, readers' actions must be continuously monitored, which introduces some privacy problems. In practice, it is difficult to implement the collaborative approach.

2.2.3 Citation Analysis Technique

While some search engines use context analysis, others use citation analysis. The citation database, CiteSeer, uses this technique to identify references relevant to the work in progress.

In Gipp's research [13], the authors illustrate citation analysis by identifying relevant references through four approaches: cited by, reference list, bibliographic coupling and co-citation analysis.

1. The cited by approach considers a reference relevant if it cites the input document (Documents A and B in Figure 2.2).

2. The reference list approach considers a reference relevant if the input document cites it (Documents C and D in Figure 2.2).

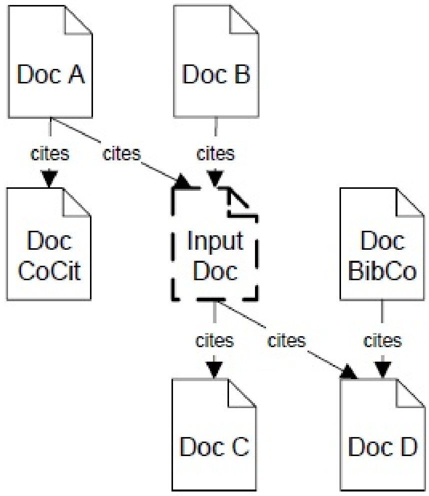


Figure 2.2: Cited by, reference list, bibliographic coupling, and co-citation approaches

3. The bibliographic coupling approach considers a reference relevant if it cites the same article(s) as the input document (Document BibCo in Figure 2.2).

4. The co-citation analysis approach considers a reference relevant if it is cited by references that also cite the input document (Document CoCit in Figure 2.2).

Citation analysis has some limitations. For example, it cannot distinguish between homographs (authors with identical names). As a result, citation analysis sometimes cannot assign a research paper to its correct author [20]. Also, irrelevant items tend to find their way in reference lists because of the *Matthew Effect*[1] , *self citations*[2] , *citation circles*[3] , and *ceremonial citations*[4] [13]. In addition, citation databases do not have the capacity to contain all the references returned by the search.

In practice, authors seldom use just one method of paper recommendation. Instead, they use

[1]The Matthew Effect describes the fact that frequently cited publications are more likely to be cited just because the author believes that well-known references should be included [21].

[2]Sometimes, self-citations are made to promote the author's other publications even though they are irrelevant [28].

[3]Citation circles occur when citations were made to promote the work of others, even though they are pointless or irrelevant [12].

[4]Ceremonial citations are citations that were used even though the author did not read the cited publication. This sometimes happens in the academic field [20].

a combined or hybrid approach of the three techniques.

2.3 Practical Solutions in Paper Recommendation System

Authors can use many existing tools for inserting citations in their work.

Finding relevant references without using any assistant tool is a time-consuming and tedious task. Authors not only spend a lot of time searching for relevant references, they also have to review them before they can manage them appropriately. Switching between writing and searching for relevant references is always disruptive. Authors find that sometimes it is easier to concentrate on reviewing and comparing references once they start searching for them instead of returning to the actual writing.

Existing practical solutions can be divided into two parts: interface evolvement and recommending technique evolvement.

2.3.1 Interface Evolvement

Three studies, from which some ideas of our system are based, were chosen to illustrate interface evolvement.

CiteSense [34] helps authors review related literature through search, selection, organization and comprehension. It also provides reference and citers' information.

Figure 2.3 shows the overview of the reference and citation information in CiteSense [34]. Panel 1 shows the paper, Panel 2 lists references cited in the paper and Panel 3 displays the citers of the paper.

Making sense of relevant literature while simultaneously searching for information is a com-

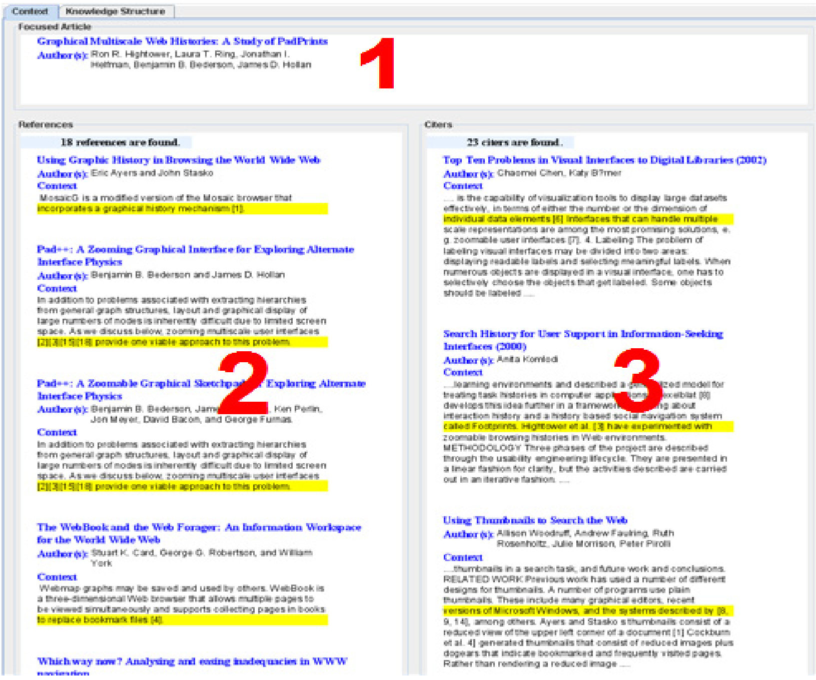


Figure 2.3: Overview of reference and citation information

plicated task. CiteSense [34] provides notes (i.e., comments about the cited content) from other sources that cited the paper. It also allows users to manage references in a separate panel.

CiteSense [34] only deals with the review of literature and lacks an editing function. Babaian et al. [4] developed Writer's Aid, an integrated system of writing and searching. Using AI planning techniques, Writer's Aid helps an author identify and insert citation marks and automatically find and save highly relevant papers and their associated bibliographic information from various online sources.

Figure 2.4 shows a snapshot of Writer's Aid. The Emacs window in the middle shows a set of citations the user has entered in his document. The body of the citation command displays the status of the searches, the first of which is completed. The window in front shows the list of references from one of the incomplete searches, while the window at the back shows the first reference from that list.

Writer's Aid [4] seamlessly integrates the search and selection of papers for citation while a

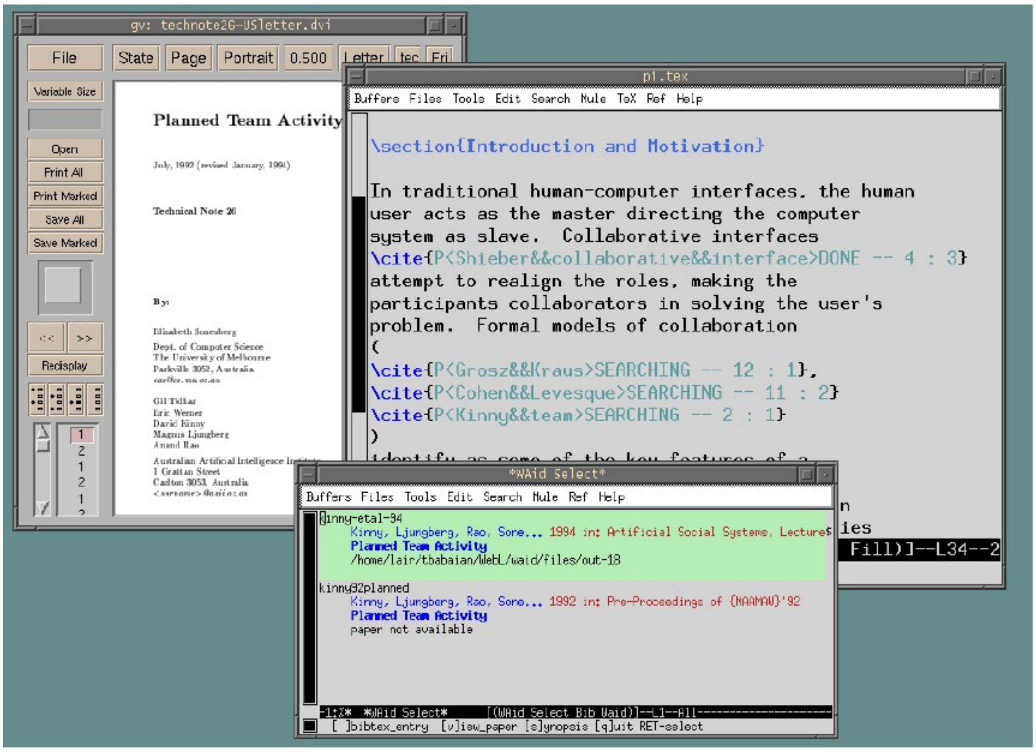


Figure 2.4: A screenshot of Writer's Aid

user is writing. However, it does not eliminate the distraction from writing since the user must specify the search terms manually when he enters a citation command.

Twidale et al. [32] claimed that the distinct activities of scholarly writing that are done in a digital library (information search, citing information and writing) can be more tightly integrated into a more spiral-like approach.

During the writing process, the content in the document constantly changes. Their system, PIRA, recontextualizes the search by generating search criteria from the changing text. This feature is also included in our system.

In PIRA, a user can switch between writing and searching and reintegrate the information into his ongoing work. Figure 2.5 shows PIRA's main display. The recontextualizing feature is not as intelligent as we expected because users have to manually specify which of the suggested terms should be included in the actual search.

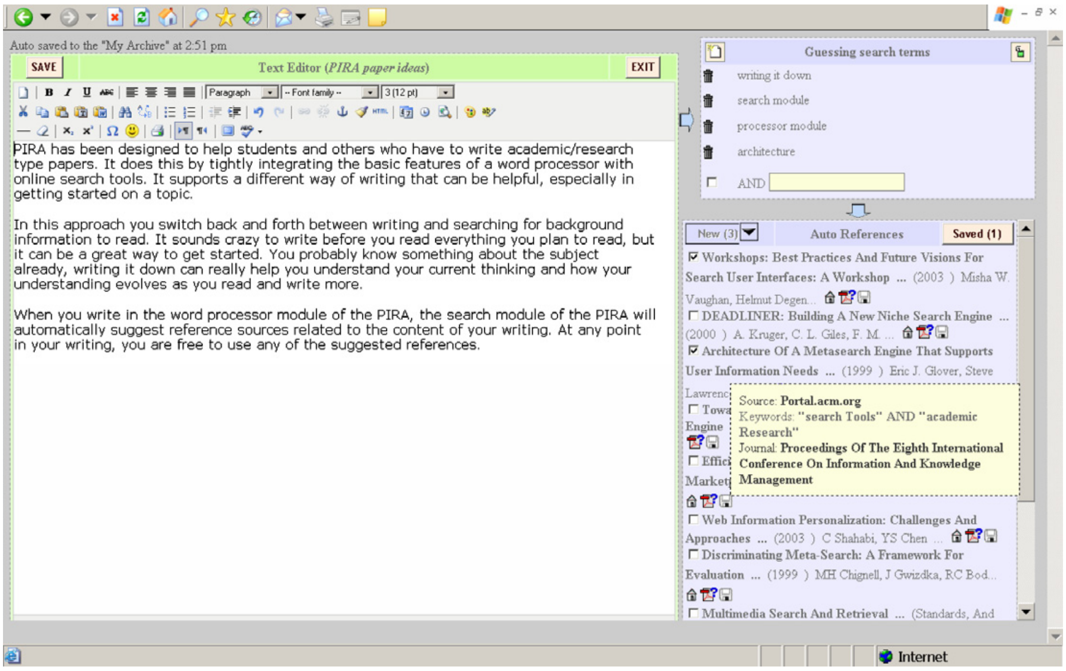


Figure 2.5: PIRA's main display showing the integration of writing and searching

2.3.2 Recommending Technique Evolvement

Different recommending techniques will lead various interfaces to display search results to users. The usual method for searching relevant resources is based on keywords or on content analysis. Apart from keywords or search terms, recommendations can also be generated using other inputs.

Woodruff et al. [33] presented a model for recommendation that uses documents instead of keywords as search criteria. Taking advantage of extensive information available in one or more documents the user has read, they used spreading activation, a mathematical technique for determining the relatedness of items based on their degree of association [2]. Recommending further reading this way enhanced the user experience in reading digital books online.

Han et al. [14] designed a rule-based agent system and a multi-agent system to autonomously find specific computer science publications on the Web. Referring to a conceptual graph of Web pages, they use heuristic knowledge to determine likely locations for citations.

The resulting recommendations are unsatisfactory. Most of them have to be refined using other techniques.

Analysis of User Type

McNee et al. [19] argued that a deeper understanding of users and their research needs results in better recommendations. They improved the quality of their recommendation system through detailed analysis of different types of users and the tasks involved in their writing. This method serves as a good guideline for developing our system's back-end.

Index Technique

There are various index techniques for querying scientific literature. Research papers are usually indexed by keyword.

The technique Bradshaw et al. [6] developed indexed research articles based on the way they are described when cited in other papers. Craven [9] mentioned automatic abstracting methods as another way of indexing research articles. This involves generating abstracts through a hybrid method that uses human effort and various computerized tools.

The automatic generation of abstracts can be optimized. Teufel and Moens [30] pointed out that robust and high-compression abstracting can be greatly improved if the discourse structure of the text is taken into account.

Summaries about the relatedness of the current work to prior research can be used as another way of indexing. Teufel [29] studied how scientific papers are related by evaluating scientific approaches in a questioning-answering task way.

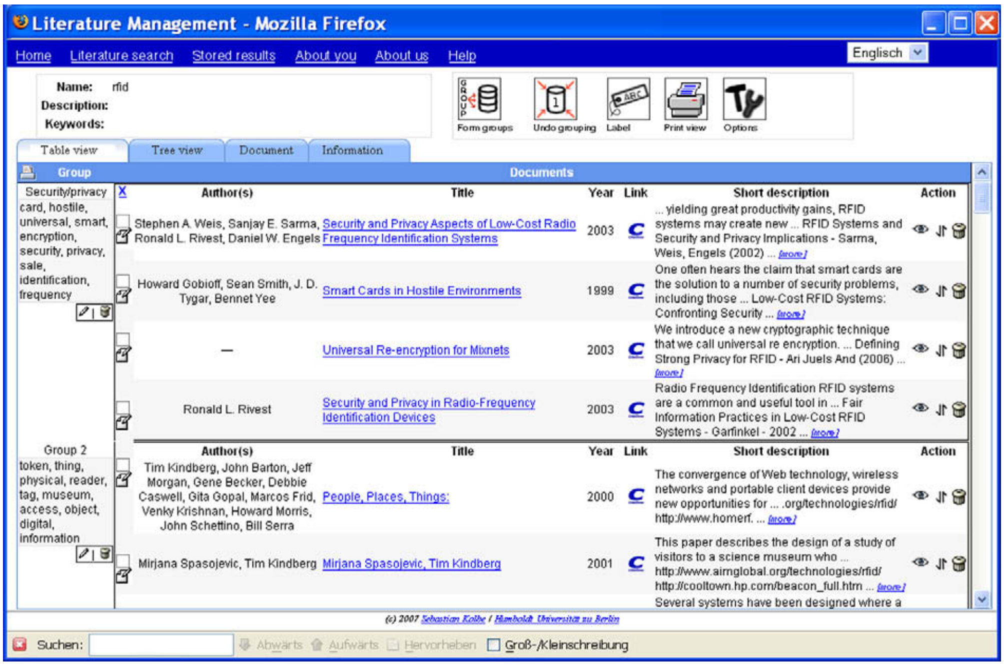


Figure 2.6: Grouping and annotation interface

Construct and Refine the Search Terms

Berendt et al. [5] proposed a system that encourages a user to actively create and refine search terms by simulating the ”reading” phases of the academic writing process (search/retrieval and sensemaking). It supports constructive clustering of literature based on search terms that users can put online for discussion.

Figure 2.6 shows the grouping and annotation interface when the user is searching for literature on “RFID.” He has already labeled the first group “security/privacy” but the second group retains its default label, “Group2.” [5]

Figure 2.7 shows a grouping result that has been put online for discussion.

Rhodes and Starner [26] argued that sometimes the recommendation system cannot help when the user does not remember enough to be able to ask a question, or does not know what to ask when querying. They designed Remembrance Agent, which performs an associative form

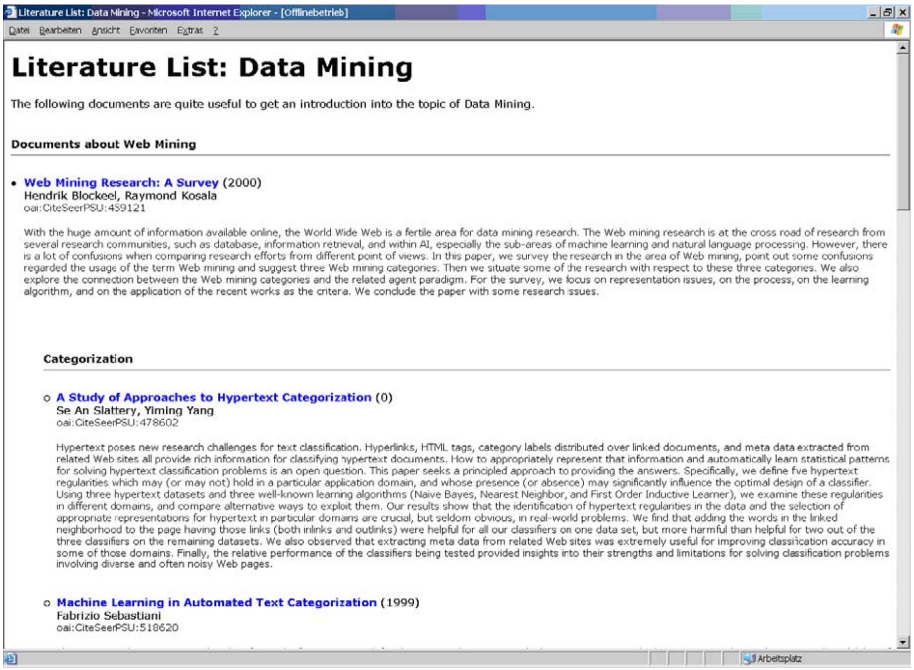


Figure 2.7: Clusters of literature published for discussion

of recall by continuously displaying information that might be relevant to the user's current content.

2.4 Summary of Related Work

Switching between searching for references and writing is still disruptive even with existing assistant tools. Our proposed system handles this problem well. It has a feature that allows the user to insert a citation marker after a sentence or phrase to inform the system he wants to cite a source there. The user can continue writing while the back-end system processes the marker information. When he finishes writing, he can return to the citation marker to perform tasks in the local suggestion window. This kind of interaction minimizes the distraction from writing.

Our system also involves more interactions through graphic interfaces and allows the user to directly manipulate the recommended citations. This makes inserting citations more intuitive and convenient.

Chapter 3

Preliminary Work

We conducted a pilot interview to examine the workflow authors followed when they are writing. During the early stage of system design, we held a paper prototype evaluation to gather user feedback on the layout design.

3.1 Pilot Interview

3.1.1 Purpose

A pilot interview was conducted to determine the typical workflow of managing citations and to gather user requirements for citation management.

3.1.2 Participants and Procedure

With the purpose of investigating the workflow of citation managing as much as we can, we recruited seven experienced academic authors for this interview: two university professors, three research fellows, and two senior PhD students. All of them, who are frequently involved in academic writing, compose more than 2 academic papers annually.

The pilot interview was divided into three stages:

1. In Stage 1, we asked the participants general questions about their writing process and the tools they used.
2. In Stage 2, we referred to one paper that each participant had written recently to ask specific questions about how they managed each of the citations.
3. In Stage 3, we asked them how they use the system that can automatically suggest reference to search for the relevant citations and download those information. We also asked several questions to determine the process they used for searching relevant citations.

All participants shared their citation strategies during the interview.

3.1.3 Results

Stage 1: The Writing Process

All participants use LaTeX, which confirms its ease of use as an editing tool.

They follow one of two types of writing process. In one process the framework of the paper (e.g., abstract, introduction, related work, experiments, etc.) is first defined before each section is filled in. In the other process, the main chapters in the abstract are defined, the details of each chapter are expounded, then the introduction is written and the abstract is refined.

Almost each participant has his own citation management process. Citations can be classified under three categories: known, roughly known, and unknown. Most of the participants in our interview often prefer to cite sources they have read before. Since they know these references, they only take a few seconds to add the citations and are not distracted from their writing.

When they are not sure about which reference to cite or whether they should put a citation at some place in the document, they insert markers such as “[ ]” or “cite( )” and continue their writing. They usually come back to deal with the citation marker after they have finished writing a section or when they are tired of writing. This prevents their flow of thought from being interrupted.

When participants know a reference well, they insert the citation directly from their local bibliographic information database in LaTeX, and link to the corresponding BibTex file that is already stored in the local database. Otherwise, they go online to search for the corresponding BibTex file, copy it into their local database and finally insert the citation from there.

For roughly known or unknown citations, they browse websites of conferences or proceedings, or perform keyword search in search engines such as Google Scholar. After searching, reading and deciding which source to cite, they obtain its BibTex file, copy it into the local database and insert the citation.

Stage 2: Managing Citations

Each participant worked with a paper they recently wrote to simulate tasks in the paper prototype.

All participants except one were familiar with all the citations in their paper. These participants had already finished reviewing literature before they started writing. They cited sources directly while they wrote.

The lone participant who was not familiar with all 31 citations in his paper has 13 well known, 12 roughly known, and 6 unknown citations before he started writing. Of the 12 roughly known citations, he knows the year and conference information of 2, the author information of 8, and has only a little impression of the remaining 2.

The participants identified the inconveniences they encountered in their use of current editing tools. Below are their suggested improvements:

1. The PDF link to the original paper and the HTML link to the author's webpage should be provided. The webpage is a useful reference as it may contain the latest work on the subject that the researcher is unaware of.

2. When the system suggests a reference, the user should be able to view the paragraphs or sections that other authors have cited.

3. Only the abstract, introduction, and conclusion should be displayed in the system since these are the sections an author reads when deciding whether or not a source is relevant.

4. It is useful to show the comments and analyses other authors have made on the cited reference

5. The system should recommend recently published papers that are relevant to the author's work.

6. It will be useful if the system could retrieve similar sentences based on the existing content and display the relevant references based on them.

7. A priority value should be assigned to each citation when there are too many to choose from. The author could then sort the suggested citations and delete the ones with low priority values. The system should give higher priority to papers in the local database since the author is already familiar with them.

Stage 3: Searching for Related References

Most of the participants use Google Scholar to find related work, using different keywords as search terms. They also browse conference proceeding websites to check for recently published papers. Some also follow the forward and backward chaining links provided by Google Scholar.

The forward chaining link shows the papers which cited a particular work and the backward chaining link shows the references that work cited. Often, participats only scan a resource's abstract to determine whether or not it is relevant. If so, they download its PDF file and read it throughly.

When the participants find the source they want to cite, they download its bibliographic information into their local database in LaTeX or use other tools such as CiteULike [8] to help them manage the bibliographic information.

Based on these responses, we asked the participants whether they would use a system if it had the following features:

1. A function that informs the system that the user wants to insert a citation at a certain location, and prompts it to recommend references based on the content that has been written so far. The results should be displayed so that it is easy for the user to scan, read, and select.
2. A function that allows the user to specify the content from which the system will base the citation search, and to be able to manage those citations intuitively.
3. A suggestion list that helps the user recall sources he has read before but cannot remember clearly.

All of them confirmed that such a system would be useful if it has a high accuracy rate in suggesting references relevant to the content they have written so far.

3.1.4 Summary

The participants immediately insert a citation if they know the reference they want to cite. When they are not sure, they tend to mark the places where citations are needed and come back to deal with them later.

While most of the participants are familiar with all the citations in their papers, some participants worked with sources that they initially did not know well. It is useful to show the abstract, introduction and conclusion section for a suggested reference as authors often scan these three sections to decide whether it is relevant to their work.

Suggesting citations based on content that has been written so far is a feature that all the participants find promising. This will be incorporated in ActiveCite.

Most people use Google Scholar to search for references. They often follow the citer and reference links to check whether a source is relevant to their topic of research. Unfortunately, they are not always satisfied with the search results since the relevance is not so high. A more accurate citation suggestion tool is required.

3.2 Paper Prototype Evaluation

3.2.1 Purpose

We conducted a paper prototype evaluation early in our design process to confirm that users understand the value of a citation suggestion system. We also wanted to have a better understanding of specific user requirements that are not addressed by traditional tools.

3.2.2 Participants and Procedure

Three participants (different from those in the pilot interview) were recruited for the paper prototype evaluation: one university professor and two senior PhD students. Academic staff and graduate students are good representatives of our target users because they are frequently involved in academic writing and citation management tasks.

One observer and one recorder conducted the paper prototype evaluation. The observer explained how ActiveCite worked while the recorder took down the participants' actions and comments. The observer also played the role of the computer, simulating the system at work by responding to users' actions.

All three participants were each asked to bring a draft of a paper they recently wrote. They were observed as they went through their drafts and performed citation management tasks using the paper prototype.

Afterwards, each participant completed a questionnaire on whether the system was intuitive to use and helpful in managing citations. It also asked if they would consider using the system if it were developed and made available.

3.2.3 Results

According to one participant, the layout design of the global suggestion window (Figure 3.1) and the local suggestion window (Figure 3.2) is not intuitive. He suggested a feature that shows all the sources that cite the recommended reference as well as the sources that the recommended reference cited. Another important suggestion is to provide a function that shows content from the recommended reference that other authors have cited, analyzed and commented on. This participant often searches for this kind of information during the course of his writing.

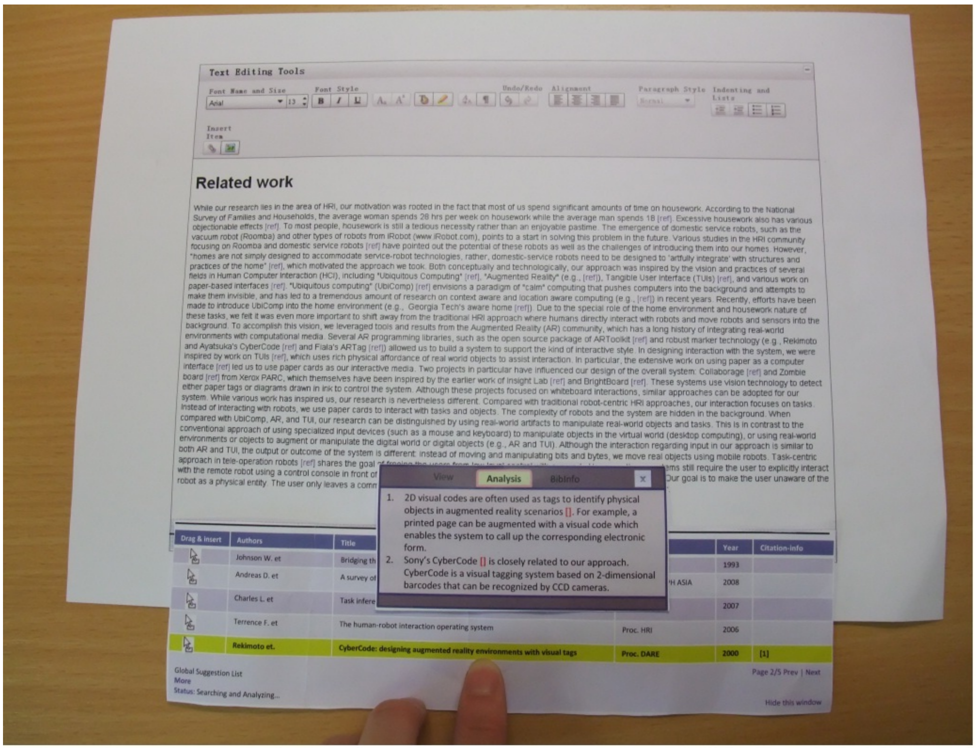


Figure 3.1: The global suggestion window of the paper prototype is the figure at the bottom

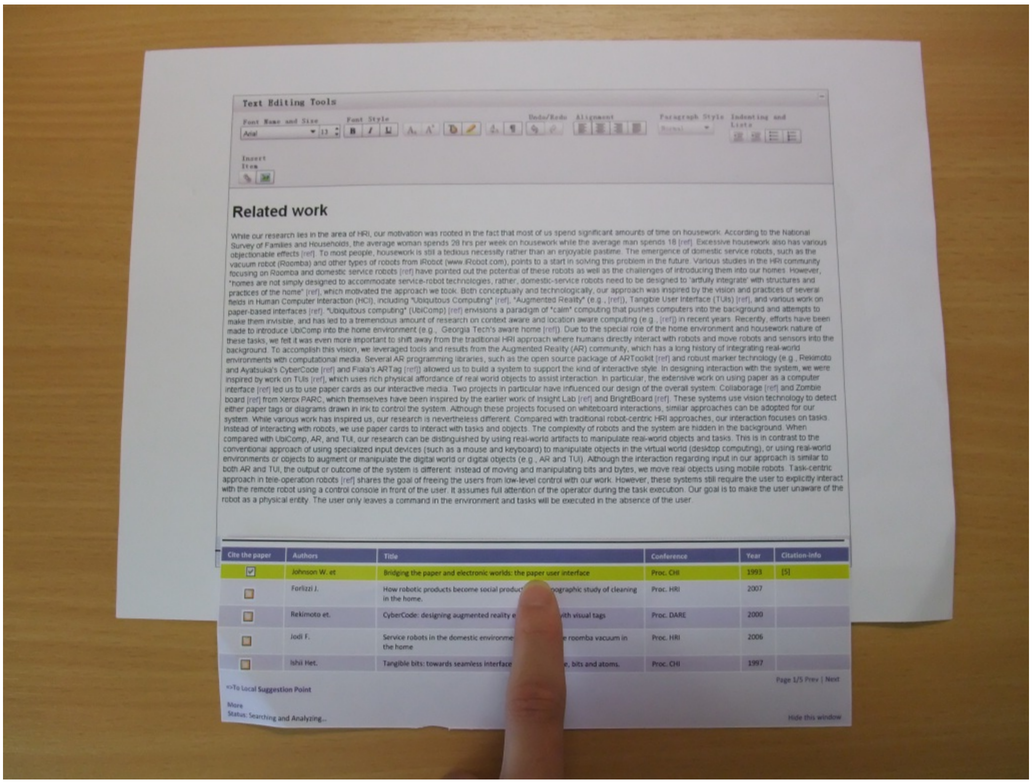
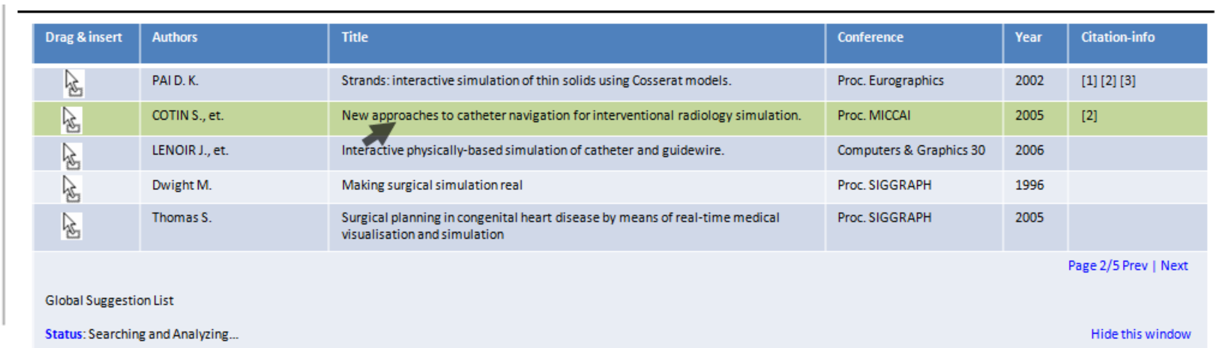
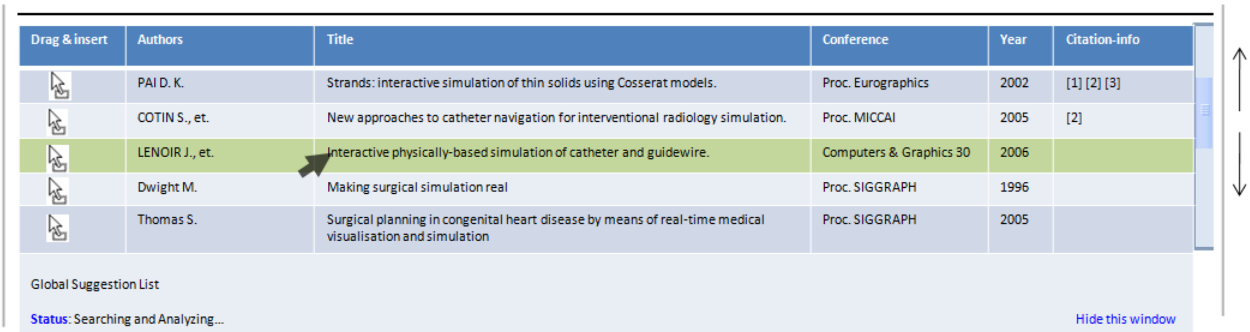


Figure 3.2: The local suggestion window of the paper prototype is the figure at the bottom



Drag & insert	Authors	Title	Conference	Year	Citation-info
	PAI D. K.	Strands: interactive simulation of thin solids using Cosserat models.	Proc. Eurographics	2002	[1] [2] [3]
	COTIN S., et.	New approaches to catheter navigation for interventional radiology simulation.	Proc. MICCAI	2005	[2]
	LENOIR J., et.	Interactive physically-based simulation of catheter and guidewire.	Computers & Graphics 30	2006	
	Dwight M.	Making surgical simulation real	Proc. SIGGRAPH	1996	
	Thomas S.	Surgical planning in congenital heart disease by means of real-time medical visualisation and simulation	Proc. SIGGRAPH	2005	

Figure 3.3: Scan the suggested papers by clicking previous/next page hyperlink



Drag & insert	Authors	Title	Conference	Year	Citation-info
	PAI D. K.	Strands: interactive simulation of thin solids using Cosserat models.	Proc. Eurographics	2002	[1] [2] [3]
	COTIN S., et.	New approaches to catheter navigation for interventional radiology simulation.	Proc. MICCAI	2005	[2]
	LENOIR J., et.	Interactive physically-based simulation of catheter and guidewire.	Computers & Graphics 30	2006	
	Dwight M.	Making surgical simulation real	Proc. SIGGRAPH	1996	
	Thomas S.	Surgical planning in congenital heart disease by means of real-time medical visualisation and simulation	Proc. SIGGRAPH	2005	

Figure 3.4: Scan the suggested papers using vertical scrollbar

Another participant shared that sometimes he cannot remember the author's name or the source's title correctly. The system would greatly improve his workflow if it included auto-complete or auto-correction features for the author's name or the source's title. He also commented that the system should use a more intuitive way of asking whether the user wants to add a citation when it finds resources that are related to the text in a given location in the document. This is better than having the system wait for the user to explicitly insert a citation marker every time.

After this round of paper prototype evaluation, some features of the paper prototype interface were changed. For instance, the suggested papers in the suggestion window can be scanned through using the vertical scrollbar instead of by clicking the hyperlink of previous or next page(changes shown in the Figure 3.3 and Figure 3.4)

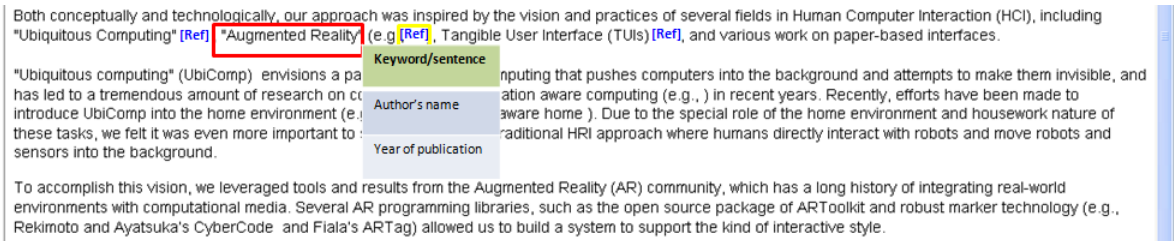


Figure 3.5: Figure 1 for auto-complete function

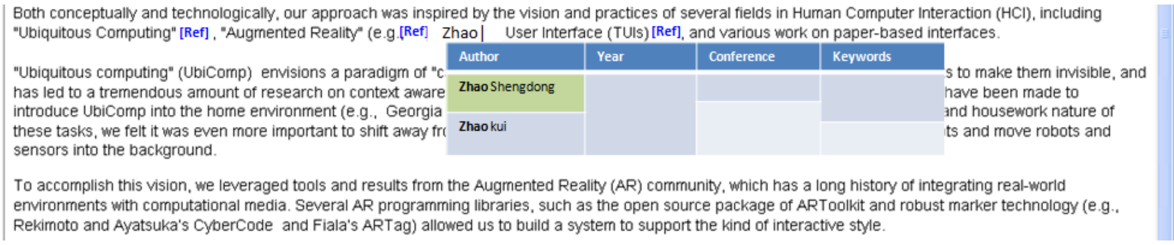


Figure 3.6: Figure 2 for auto-complete function

In addition, auto-complete function was added to the paper prototype interface. From figure 3.5 to figure 3.9, it shows some user scenario of auto-complete function while searching relevant papers for ”Augmented Reality”.

As required by one participant, the function of viewing pdf file of suggested paper was embedded into the paper prototype interface(shown in Figure 3.10), though we found it is difficult to download the corresponding pdf file and open it within our platform during the implementation.

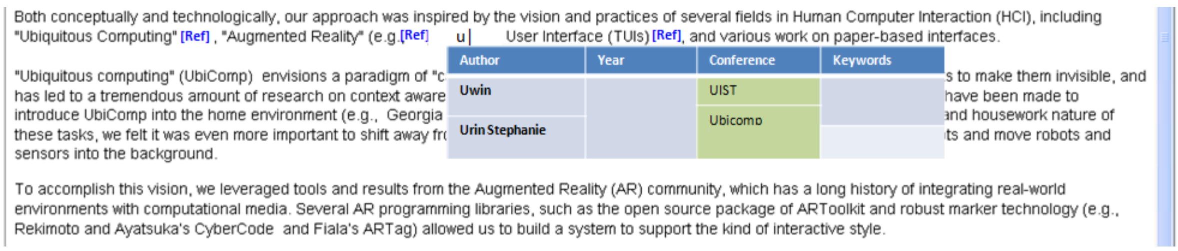


Figure 3.7: Figure 3 for auto-complete function

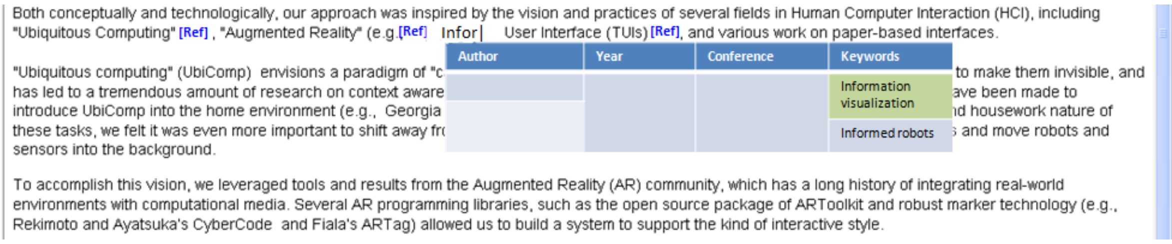


Figure 3.8: Figure 4 for auto-complete function

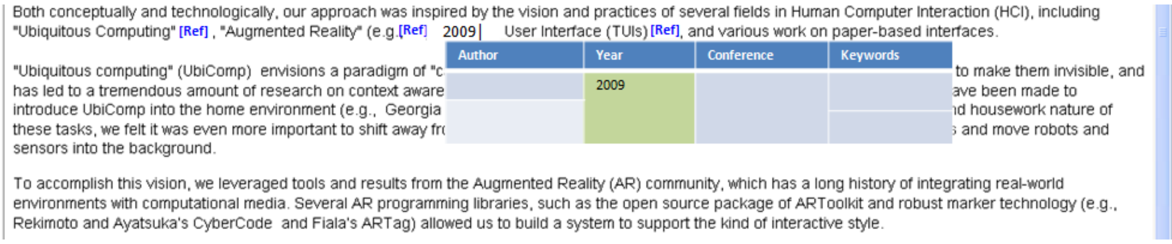


Figure 3.9: Figure 5 for auto-complete function

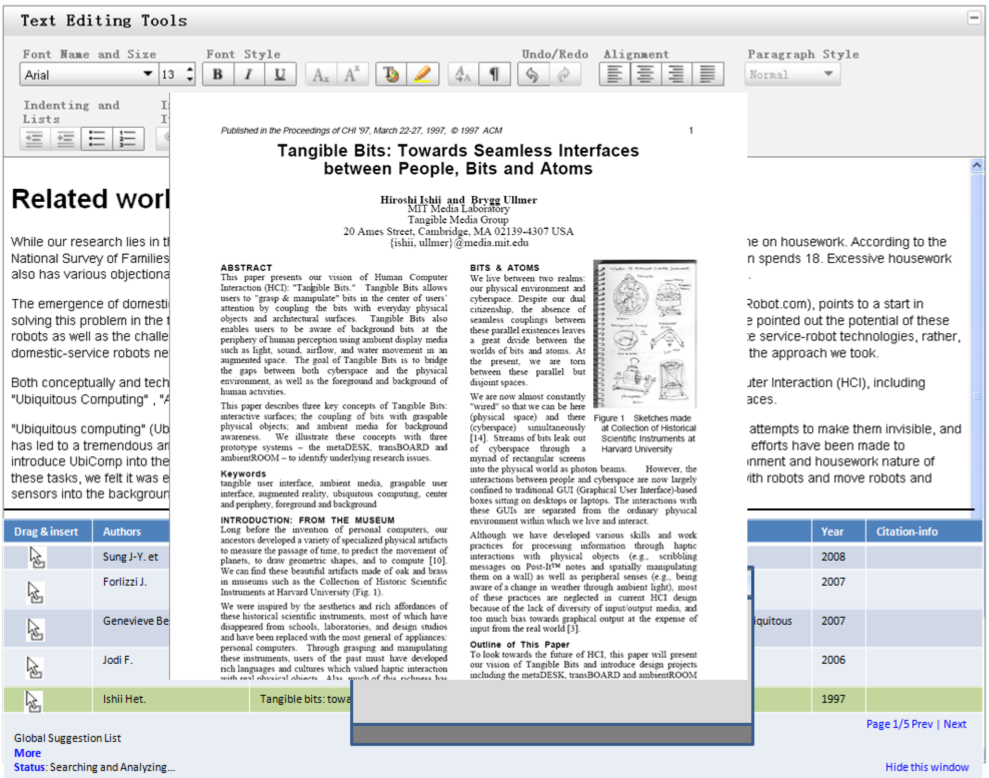


Figure 3.10: Figure 5 for auto-complete function

3.2.4 Summary

The participants identified the limitations of our preliminary design, which prompted us to re-examine aspects of the task workflow that we had overlooked. We acquired many significant user requirements from the session of the paper prototype evaluation. Our system has considerably evolved since.

Chapter 4

User Scenario

In this chapter, a complete user scenario is created to illustrate what a user can do with our system prototype.

Sam, a senior research scientist, uses ActiveCite to write an academic paper.

4.1 Using the Global Suggestion Window

While Sam is writing, the system automatically retrieves the references that are highly relevant to the content he has so far and displays them in the global suggestion window. If he does not want to read the suggested sources yet so that he can focus on writing, he can hide this window by clicking the Hide button located at the top-left side of the global suggestion window.

When Sam is ready to cite a reference, he clicks the Show button at the bottom of the editing panel to bring up the global suggestion window. He browses through the suggestion list and reads the details (e.g., abstract, citer list, reference list) of a particular reference by clicking its title.

Each reference has a corresponding icon in the Drag and Insert column of the global suggestion window. Sam inserts a citation by dragging and dropping its icon to any location in his document.

4.2 Using the Local Suggestion Window

Sam needs to use the local suggestion window if:

1. He cannot find the reference he wants to cite from the global suggestion window;
2. He would like to insert a citation at his current position in the document, but could not remember specific information about the reference. Neither does he have its bibliographic information in his local BibTex database; or
3. He wants to check whether there are other references related to the content he has just written.

By typing "[ref]" in the document, Sam tells the system to retrieve relevant references for him to confirm. The recommended references are based on the phrase before the [ref] marker or the text between a preceding marker and the [ref] marker.

After Sam has typed a "[ref]" marker, it is highlighted in red to indicate that the system is searching for relevant references in the back-end while Sam proceeds with his writing. When the search is complete, the red highlight changes to blue.

Sam may click the blue markers any time to view the suggestion list, which is displayed in a pop-up window. Once he finds a reference he wants to cite, he can insert the citation in the blue [ref] marker by ticking the reference's corresponding checkbox.

Sam can view details such as the abstract, citer list, reference list, and analysis (the content that other authors who cited the reference commented on) to help him decide whether the reference is relevant to his work.

Sam can refine the list of suggested references by manually entering other criteria. For instance, he can specify the year of publication, authors' names, keywords or sentences.

When Sam is ready to review the citations he has inserted, he can look at the Citation-Info column in either the global suggestion window or the local suggestion window. This column shows which references in the list are cited in the document. Clicking the citation info column of a reference pops up a window that shows the position(s) in the document where it is cited. Sam could add more citations at these locations (i.e., when a statement needs to be supported by a recently published reference) by entering appropriate criteria at the tab for refining.

Sam can view a formatted list of all his citations by clicking Current Reference under the File menu, but he cannot add or delete citations from this view.

Because the system recommends references based on content the user has written so far, it gives the advantage of helping the author learn about other people's work in the context of his own. This is especially useful for those who are reviewing literature on a research area that requires complete coverage of previous work.

Chapter 5

Prototype System

5.1 System Architecture

ActiveCite's front-end is a general editing tool that is used for editing the content of the document. Its back-end is a recommendation system that performs the search and retrieves relevant references. The data is communicated between the front-end and the back-end through the web service that acts as an intermediate layer. Figure 5.1 shows the prototype's architecture.

Queries can be made from the editing interface in order to find a reference that the user can cite. There are two types of queries. The local query is generated explicitly when the user requests the back-end system to suggest citations. The global query, on the other hand, is generated implicitly when the system takes the whole content the user has written so far to perform a search.

Figure 5.2 shows ActiveCite's main interface. Our prototype system is divided into two main parts: the global suggestion window and the local suggestion window.

The global suggestion window lists the relevant references based on the global query while

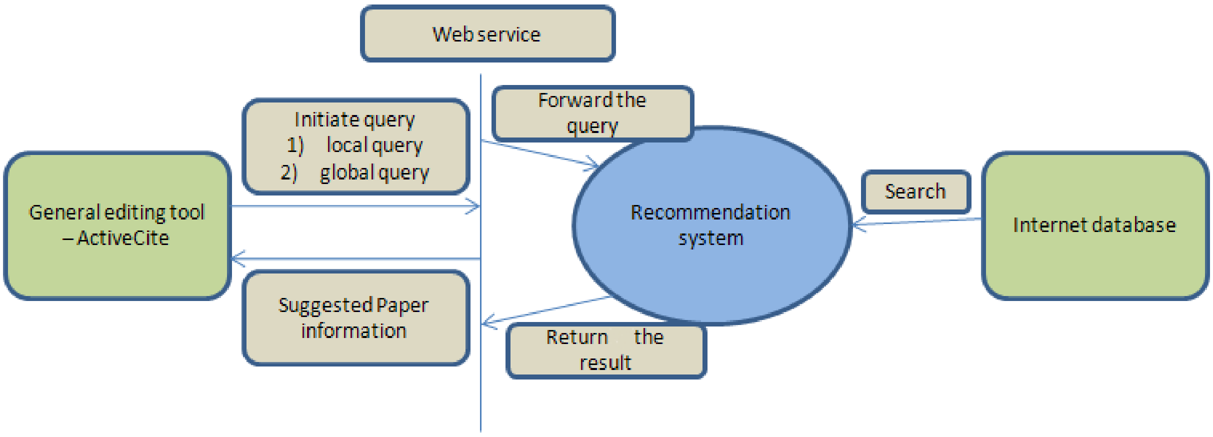


Figure 5.1: System architecture of ActiveCite

the local suggestion window lists the relevant references based on the local query.

5.2 Interaction and Visualization Techniques

5.2.1 Global Suggestion Window

The global suggestion interaction will use text mining algorithms to first analyze the content of the entire document, and then extract keywords, the author's name, year of publication and even its working title to use as search terms. These will be sent to the back-end recommendation system, which performs the search for relevant references. At the current stage of our prototype implementation, the system simply takes the entire content and uses it to automatically generate a global query.

The global suggestion window, shown in the bottom part of Figure 5.2, lists the references recommended by the system based on its analysis of the document's content. The operation is initialized when the system is run for the first time. The user can refresh the list of suggested references by clicking the Refresh button at the bottom of the global suggestion window.

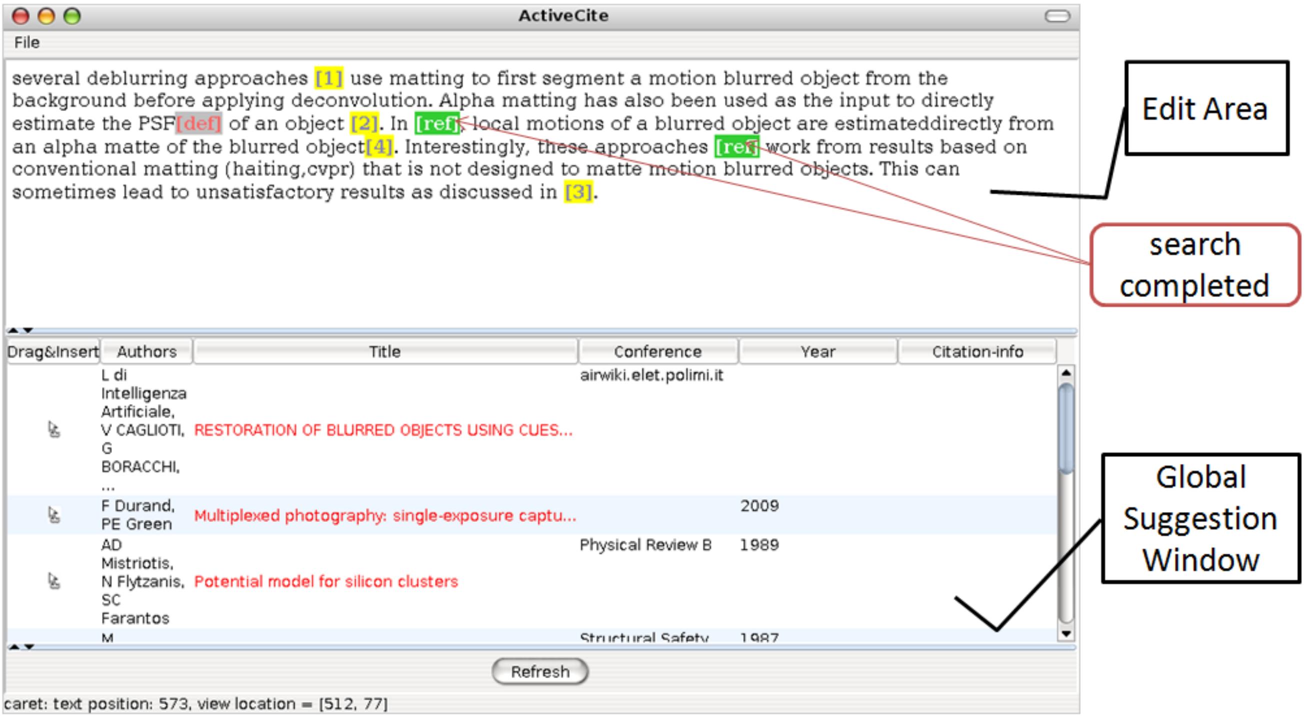


Figure 5.2: The main interface of ActiveCite

The references listed in this window are, to a certain extent, active suggestions. Once the system can intelligently decide when to automatically refresh the global suggestion list based on the changing content of the document, the list truly becomes an active suggestion.

The enlarged view of the global suggestion window (Figure 5.3) shows the reference list's column information. The user can view the details of a particular reference by clicking its title. These are displayed in a pop-up window, as shown in Figure 5.5. The user can insert a citation by dragging a reference's icon, which appears in the first column, and dropping it in the document where he wants the citation to be.

5.2.2 Local Suggestion Window

The local suggestion window (Figure 5.4) is the pop-up window that appears when the user clicks on a blue [ref] marker. It contains a list of suggested references. By entering information

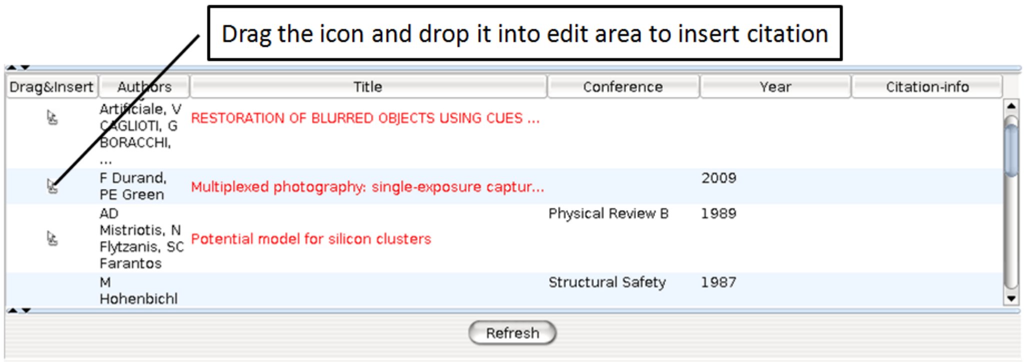


Figure 5.3: The global suggestion window

in the Contextual Search tab, the user makes a local suggestion interaction that refines the list.

The local suggestion window has several tabs: Contextual Search, Local Suggestion List, and Remove the Reference Point.

1. The Contextual Search tab shows the criterion information the user has manually entered for the [ref] marker corresponding to the pop-up window. The user can refine the search by filling up the form again and clicking the Search button.

2. The Local Suggestion List tab shows the list of suggested references associated with a specific marker. If the user he wants to cite a reference from the suggestion list, he can tick its corresponding checkbox in the Cite column. To remove a cited reference, he simply unmarks its corresponding checkbox. When the user clicks a reference's title, a window displaying its details (see Figure 5.5) pops up.

3. The Remove the Reference Point tab is where the user can remove all reference information associated with the [ref] marker.

The search terms that were used for the local query is underlined in the editing tool every time the local suggestion window pops up.

The user can click a title from either the global suggestion window or the local suggestion

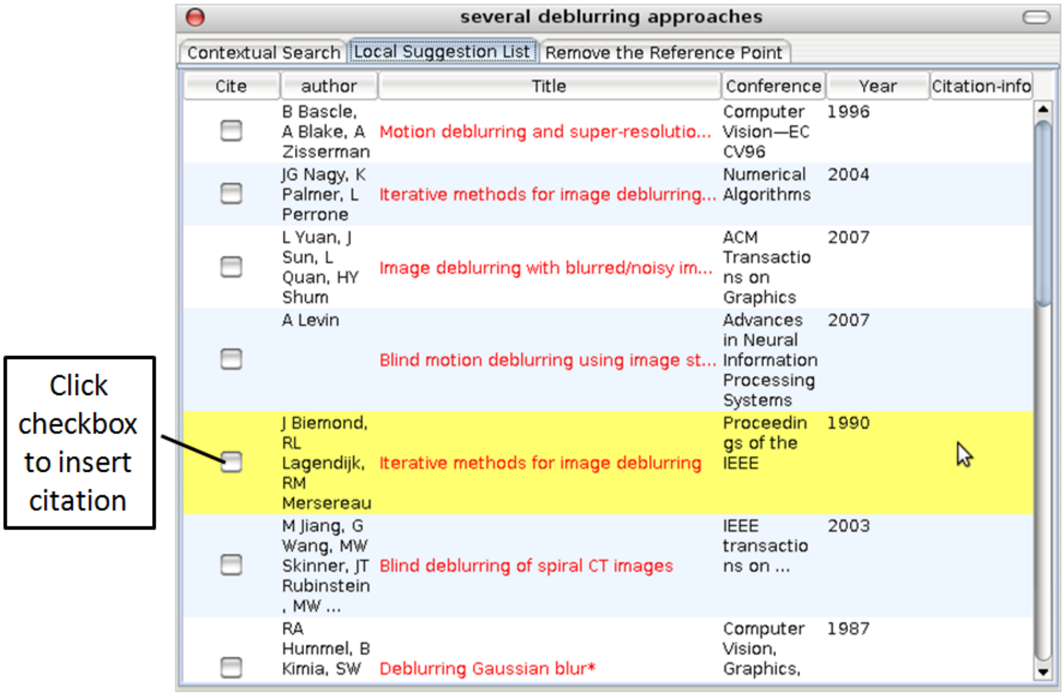


Figure 5.4: The local suggestion window, the pop-up window that appears when the user clicks the blue [ref] marker

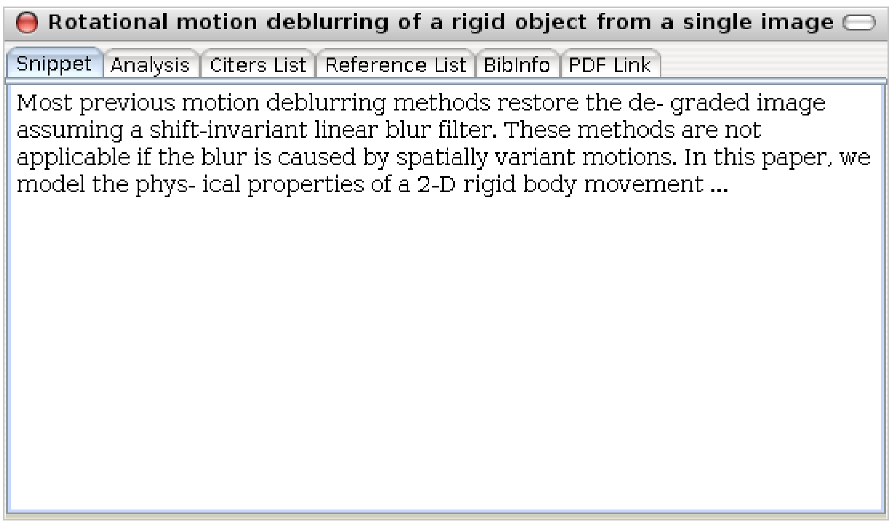


Figure 5.5: The view of a reference's abstract, which opens when the user clicks the title of a reference

window to view a reference's details. Figure 5.5 shows the details of a suggested reference. There are six different tabs in this view:

1. The Abstract tab (Figure 5.5) shows the reference's abstract. This information gives the user a sense of what the paper is about.
2. The Analysis tab (Figure 5.6) shows information on what other authors who also cited this reference wrote in their papers.
3. The Citers List tab (Figure 5.7) shows forward chaining. It displays a list of papers that cite the suggested reference.
4. The Reference List tab (Figure 5.8) shows backward chaining. It lists all the references cited by the suggested reference.
5. The BibInfo tab (Figure 5.9) shows the suggested reference's general bibliographic information, which includes the title, author, year and place of publication.
6. The PDF Link tab (Figure 5.10) provides a link to the suggested reference's PDF file, if available. The user can read the full PDF file in our integrated platform.

Both the local suggestion window and global suggestion window have a Citation-Info column, which shows whether a suggested reference has already been cited, and where in the document its citation is.

ActiveCite helps the user find relevant references in three ways: if he already knows the reference he is citing, the system helps him confirm it. If he only has partial information, the system helps him remember. If he doesn't know what to cite, the system expands his knowledge on his research topic by recommending related references to support his work.

In Figure 5.11, it shows the full picture of our prototype system when user inserted one citation after he/she checked abstract of the suggested paper,which is recommended based on

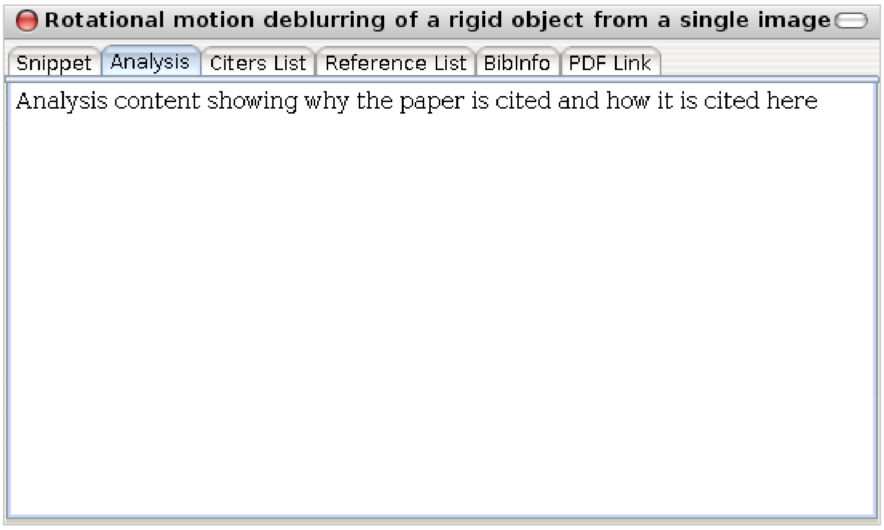


Figure 5.6: The analysis tab of the suggested reference

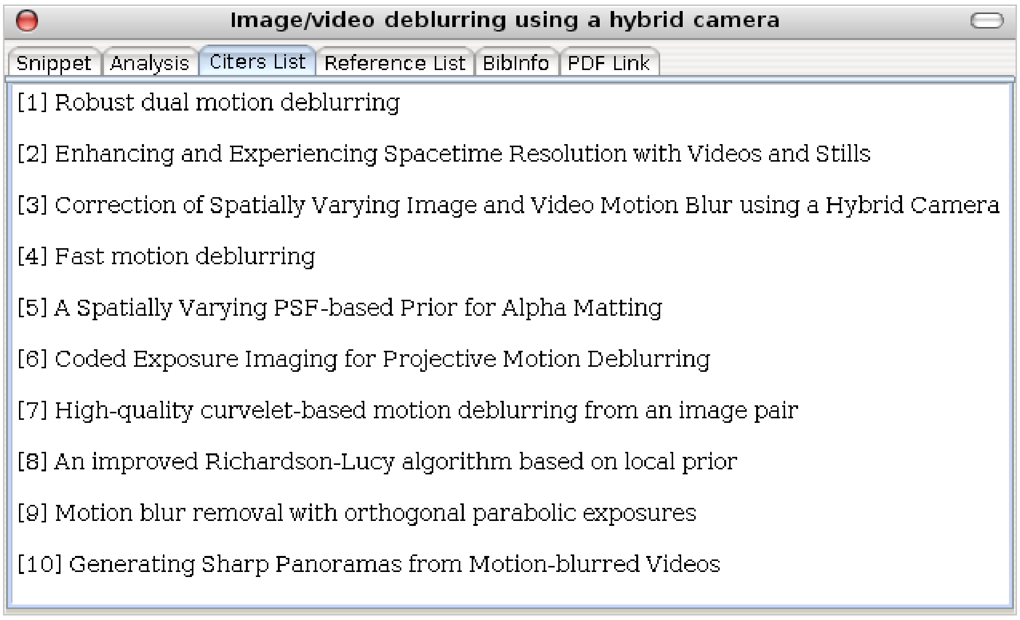


Figure 5.7: The citers list of the suggested reference (forward chaining)

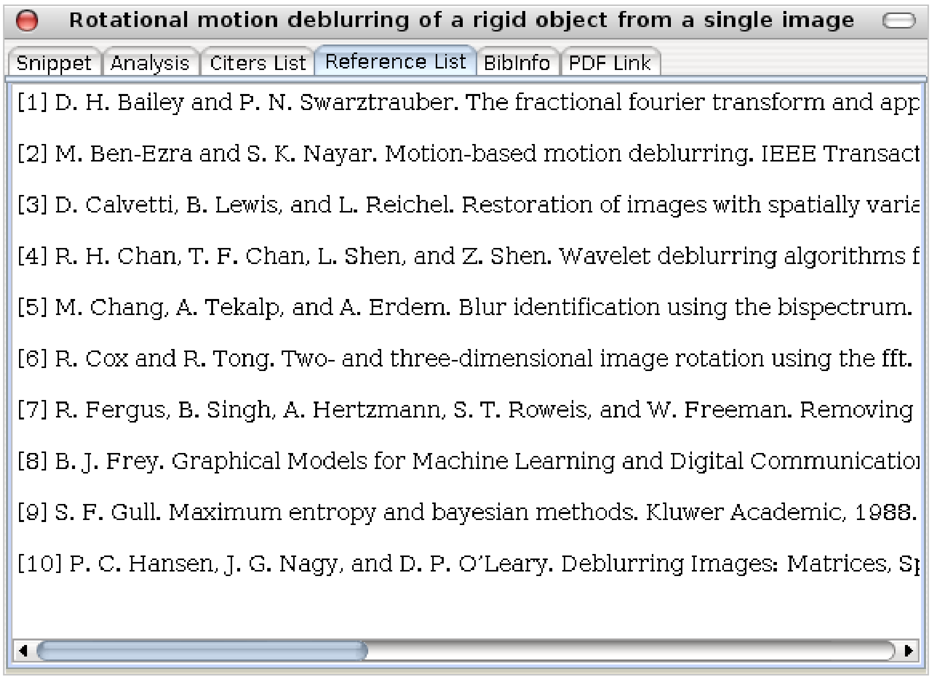


Figure 5.8: The reference list for the suggested reference (backward chaining)

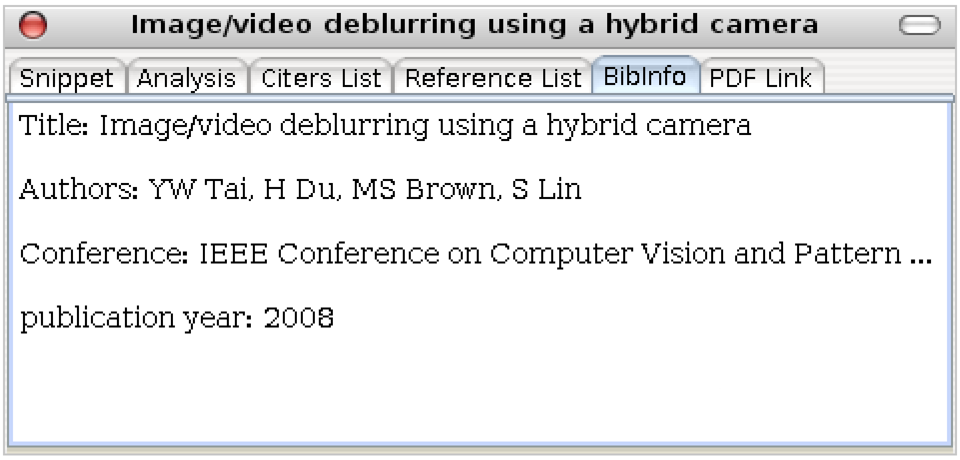


Figure 5.9: The bibliographic information of the suggested reference

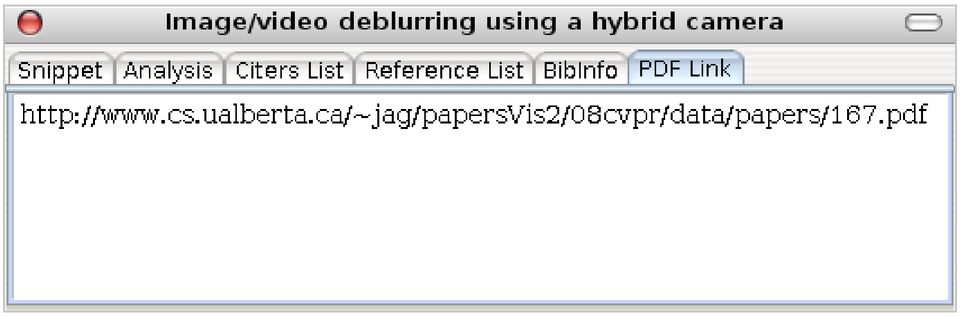


Figure 5.10: The link to the PDF file of the suggested reference

the underlying sentence before the citation marker.

5.3 Implementation

ActiveCite was developed on a desktop PC with Intel(R) Core(TM)2 Quad Q9550 @ 2.83GHz CPU and 4GB RAM running the Ubuntu 9.04 Operating System. The prototype was written in Java. The recommendation system in the back-end was developed in Perl.

To test ActiveCite's interaction with a real database, Google Scholar (GS) was used as a pseudo-recommendation system. This gives the advantage of accessing information from a live database.

When the user enters a query, GS returns a structured HTML page that the system processes. The system compiles a list of top N results (currently N=20) by parsing this page and putting the information it needs into an XML file. Each result consists of standard metadata (title, author, *abstract*[1], proceeding/conference, publication year, link to PDF), and the forward chaining information, i.e., the citers' list.

Long titles and conference names are often truncated by GS. To avoid user dissatisfaction,

[1]We are currently using the GS snippet as the abstract. In the future, we will take the original abstract by parsing the PDF file, if it is available.

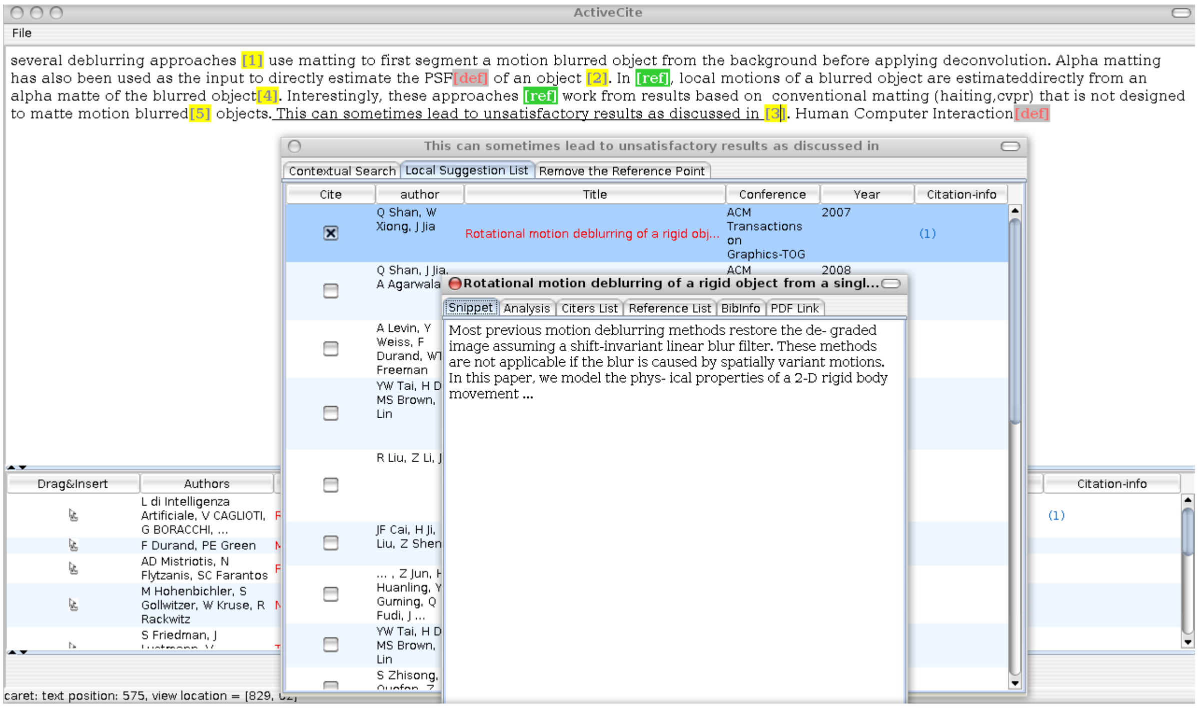


Figure 5.11: The full picture of using our prototype system

we made an effort to recover them using *Google API*[2]. We first query Google Scholar by using the truncated string to retrieve top M titles (M=10) in the resulting HTML page, and then compare these titles with the truncated string by using the standard algorithm, Longest Common Subsequence. The title most similar to the truncated string, which is the one with the highest score above t (empirically set at 0.8), is selected. This enables our system to generate a list of suggested references with complete metadata.

Following suggestions from the three university professors in the group, we added a definition search to the prototype's functions. The system uses *Wikipedia Miner API*[3] to retrieve basic definitions of terms that are entered in a search. A user can type the predefined [def] marker to get a term's definition the same way citations are acquired.

The system performs an implicit search based on the valid combination of three words preceding the [def] marker. The user can refine the search by filling in the TextField in the Refine

[2]Google SOAP Search API - http://code.google.com/apis/soapsearch/
[3]Wikipedia Miner API - http://wikipedia-miner.sourceforge.net/

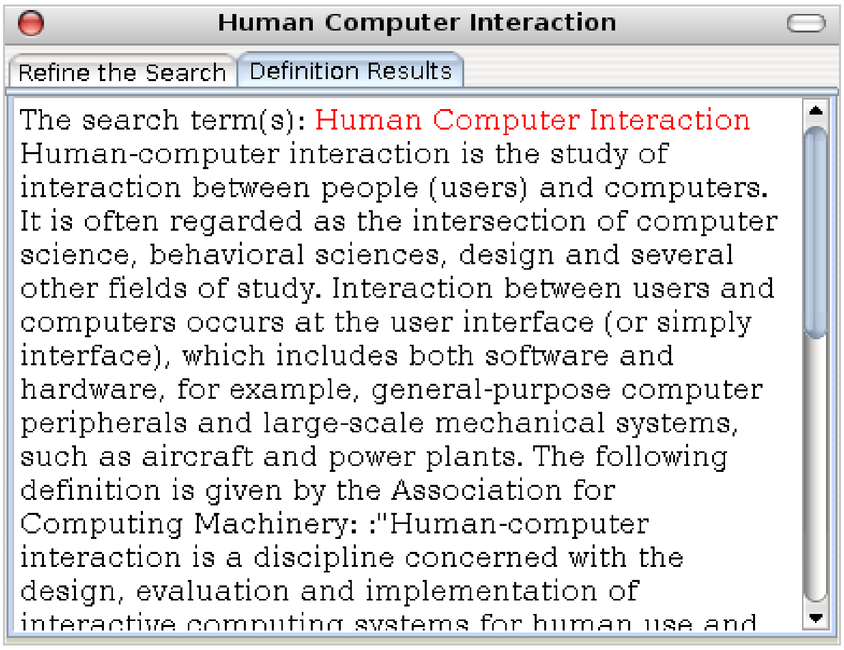


Figure 5.12: The definition window

the Search tab. Figure 5.12 shows the definition for ”Human Computer Interaction.”

Chapter 6

Initial User Evaluation

This chapter provides a discussion of the initial user evaluation we conducted, an analysis of its results, and subjective feedback from the participants.

6.1 Purpose

The purpose of the initial user evaluation is to assess the prototype system's usability and its user interface's intuitiveness. A questionnaire was given to a group of participants to measure the acceptance level of their use of the prototype.

Participants' subjective feedback identified the prototype's pros and cons as well as the functions and features that should be retained, redesigned or disregarded. These shall serve as guidelines for our future work.

6.2 Apparatus

The experiment was conducted in our laboratory and was run on a desktop computer that uses Intel(R) Core(TM)2 Quad Q9550 @ 2.83GHz CPU processor with 4GB RAM and runs on the Ubuntu 9.04 Operating System. The display is a 19" screen with a 1600 x 900 resolution.

6.3 Participants and Procedure

Five male participants (different from those in the pilot interview and in the paper prototype evaluation) took part in the initial user evaluation. All are PhD students from the Department of Computer Science with experience of academic writing, composing from 1 to 3 academic papers annually.

Each participant was asked to bring a paper they recently wrote so that they have material to work with when they use the prototype. Instructions were provided in hard copy. After we explained how to use the system prototype, we asked each participant to either rewrite one paragraph from the Related Work section of their paper or to compose a similar paragraph in the blank editing area on ActiveCite.

We observed the process each participant followed in inserting citation markers where references are needed, or in placing the definition marker beside a term he wanted a definition for.

Each participant viewed the recommended references in either the global suggestion window or the local suggestion window in order to insert citations.

After completing the exercise, participants answered a 7-point Likert scale questionnaire. They then gave their subjective feedback on the pros and cons of the prototype system.

Questionnaire Item	Mean(Std. Dev.)
Adding citations is time-consuming in an actual writing scenario	2.8(0.83666)
ActiveCite is an intuitive tool for adding citations	6(0.707107)
The definition search function is useful	5.6(0.894427)
Results based on initial search terms are accurate	3.4(1.67332)
Results based on refined search terms are accurate	6.4(0.894427)
ActiveCite helps manage citations during the writing process	6(1.732051)
ActiveCite will be used when it is fully developed	6.2(1.78854)

Table 6.1: Questionnaire responses

6.4 Results and Analysis

Given the two options of rewriting and composing, all the participants chose the rewriting task.

The results for the questionnaire are shown in the table 6.1, which demonstrates the mean values and standard deviations of subjective responses.

Questionnaire results are summarized in Table 6.1, which shows the mean values and standard deviations of subjective responses.

All participants expressed that it is time-consuming to add citations during the actual writing process (1 stands for "very time consuming" and 7 stands for "not time consuming.") This indicates that they often took a lot of time searching references, reading them, and deciding whether to cite them. Since they complete these activities before they start writing, they already know what references to cite once they begin.

The participants find the process of typing "[ref]" to indicate that a citation is needed and the interface for adding citations are intuitive. Responses have a high mean value of 6 (1 stands for "not intuitive" and 7 stands for "very intuitive.")

Although definition search is rarely used in assistant tools for academic writing, all the participants stated that it is a useful feature (1 stands for "not useful" and 7 stands for "very useful.") Participants said that they sometimes need the standard definition of a specific term (e.g., mathematics rules) instead of a simplified or informal definition recalled from memory.

Almost all the participants were not satisfied with the list of suggested references that was generated by the initial search. However, they were satisfied with the list of suggested references based on the refined search terms (e.g., authors, title, year and conference name) that they entered themselves. One participant gave 6 for satisfaction based on initial search terms because he successfully found several references that were relevant to his paper. It is worth noting that the words in his paragraph, from which the system based its search, were more specific to the topic of his paper compared to the words in the other participants' paragraphs.

All except one participant agreed that ActiveCite is very useful in managing citations. They said they would use it when it is fully developed with an improved back-end system for recommending papers and full editing features similar to MS Word or LaTeX. The participant who did not share the majority's opinion gave 3 for ActiveCite's usefulness and 3 for the willingness to use it, which resulted in high values of standard deviation for these two items. Since this participant is very familiar with all the references he wants to cite, he does not think about citations when he is writing.

6.5 Subjective Feedback

After the participants filled up the questionnaire, they were asked to give subjective feedback about ActiveCite. Several useful suggestions, which can be developed into additional features in the future, are listed as follows:

1. A feature should be developed to make it easier to manage citations that appear several times in the document. The user should, for instance, be able to drag and drop one citation sequence number from one place in the document to another.

2. Deleting citations from the reference list should enabled.

3. When the user cites a reference, ActiveCite should automatically download its PDF file and formatted bibliographic information into the local database.

4. A more advanced text mining preprocess is needed in order to define the terms for the initial search. This will improve the accuracy of the search for relevant references.

5. Combining metadata (such as author, conference name and year) with the phrase preceding the [ref] marker can improve the relevance of the search results. In the current prototype, the system finds it difficult to use words in a generic sentence as search terms.

6. The system should be able to determine which topics or categories suggested references fall under.

7. The user should be able to choose from several citation formats since different types of academic writing follow different formats.

8. The local suggestion window should pop up when the mouse hovers over the blue [ref] marker. This is better than having to click the marker before the list of suggested references can be viewed.

Most of the participants expressed that ActiveCite can make a significant and promising contribution to academic writing if its citation management feature was integrated as a plug-in in an existing editing tool (e.g., MS Word, LaTeX).

6.6 Summary

The initial user evaluation was conducted to obtain feedback on ActiveCite's usability. After the prototype's editing functions and back-end are refined, a comparison experiment between ActiveCite and a current writing platform (e.g., LaTeX plus Google Scholar or Word plus Google Scholar) may be conducted.

Chapter 7

Conclusion and Future Work

Finding and inserting citations while people are writing academic papers is not an easy task. Academic writers need useful tools to help them in this progress. We found that a lot of related work has been done. Based on our literature review, we proposed our own system, ActiveCite, with the focus on the intuitive interaction of the user interface.

ActiveCite is a tool for academic writing. It is an interactive system that aims to improve the typical workflow of adding citations by automatically recommending relevant references. And it also tries to reduce the disruption of switch between writing and searching for papers.

At its current stage of development, ActiveCite uses the Google Scholar search engine as its recommendation system. Thus, the accuracy of the search results is not very high. We will consider deploying our own recommendation system, which probably will be developed by Natural Language Process group, to increase the recommended references' relevance to the work in progress.

We recruited 7 participants for the pilot interview to investigate academic writer's workflow of writing paper. After that, 3 participants were called for the paper prototype evaluation to help us iteratively design the layout of ActiveCite. When we finished developing the prototype system, 5 participants evaluated it for the initial user evaluation.

The findings of the pilot interview, paper prototype evaluation and initial user evaluation are very encouraging and useful for our system's iterative design. Based on these findings, we set the following points as our our design goals for the current stage and future.

1. To minimize the effort to locate relevant references,
2. To help the user readily access relevant references he has read before without storing all their information beforehand, and
3. To recommend relevant references that the user has not yet come across.

For future work, we shall further refine ActiveCite's layout by making the user interface more intuitive. We shall also improve the accuracy of its recommendation system. Afterwards, we shall conduct a comparative experiment between ActiveCite and existing tools to evaluate how well it assists authors in managing citations.

Bibliography

[1] Nitin Agarwal, Ehtesham Haque, Huan Liu, and Lance Parsons. Research paper recommender systems: A subspace clustering approach. In *INTERNATIONAL CONFERENCE ON WEB-AGE INFORMATION MANAGEMENT (WAIM)*, pages 475–491, 2005.

[2] J.R. Anderson and P.L. Pirolli. Spread of activation. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 10(4):791–798, 1984.

[3] Writing Center at Yale University. Signaling sources in the body of your paper. http://www.yale.edu/bass/writing/sources/kinds/principles/signaling.html.

[4] Tamara Babaian, Barbara J. Grosz, and Stuart M. Shieber. A writer's collaborative assistant. In *IUI '02: Proceedings of the 7th international conference on Intelligent user interfaces*, pages 7–14, New York, NY, USA, 2002. ACM.

[5] B. Berendt, B. Krause, and S. Kolbe-Nusser. Intelligent scientific authoring tools: Interactive data mining for constructive uses of citation networks. *Inf. Process. Manage.*, 46(1):1–10, 2010.

[6] Shannon Bradshaw, Andrei Scheinkman, and Kristian Hammond. Guiding people to information: providing an interface to a digital library using reference as a basis for indexing. In *IUI '00: Proceedings of the 5th international conference on Intelligent user interfaces*, pages 37–43, New York, NY, USA, 2000. ACM.

[7] J. D. Bransford and J. J. Franks. Sentence memory: A constructive versus interpretive approach. *Cognitive Psychology*, 3:331–350, 1971.

[8] CiteULike. http://www.citeulike.org/.

[9] Timothy C. Craven. Human creation of abstracts with selected computer assistance tools. *Inf. Res.*, 3(4), 1998.

[10] P. Elbow. *Writing with power: techniques for mastering the writing process*. New York, NY: ACM Press, 1998.

[11] Barbara Fister. The research processes of undergraduate students. *Journal of Academic Librarianship*, 18(3):163–169, 1992.

[12] E. Garfield and A. Welljams-Dorof. Citation data: their use as quantitative indicators for science and technology evaluation and policy-making. *Science and Public Policy*, 19(5):321–327, 1992.

[13] Beel J. Gipp B. Scienstein: A research paper recommender system. *International Conference on Emerging Trends in Computing. IEEE*, pages 309–315, 2009.

[14] Yi Han, Seng Wai Loke, and Leon Sterling. Agents for citation finding on the world wide web. In *Proceedings of the Practical Application of Intelligent Agents and Multi-Agent Technology 1997 C PAAM 97*, pages 303–318, 1997.

[15] LaTeX. A document preparation system. http://www.latex-project.org/.

[16] Charles Lipson. *Doing Honest Work in College: How to Prepare Citations, Avoid Plagiarism, and Achieve Real Academic Success*. University Of Chicago Press, October 2004.

[17] A. Deshpande Mari, L. Boves and Noz. A proactive recommendation system for writing: helping without disrupting. pages 89–95, New York, NY, USA: ACM, 2007. In ECCE '07: Proceedings of the 14th European conference on Cognitive ergonomics.

[18] R.M. May. The scientific wealth of nations. *Science*, 275(5301):793–796, 1997.

[19] Sean M. McNee, John Riedl, and Joseph A. Konstan. Making recommendations better: an analytic model for human-recommender interaction. In *CHI '06: CHI '06 extended abstracts on Human factors in computing systems*, pages 1103–1108, New York, NY, USA, 2006. ACM.

[20] L.I. Meho. The rise and rise of citation analysis. *Physics World*, 2006.

[21] R.K Merton. The matthew effect in science. *Science*, 159(3810):56–63, 1968.

[22] Stuart E. Middleton, Nigel R. Shadbolt, and David C. De Roure. Ontological user profiling in recommender systems. *ACM Trans. Inf. Syst.*, 22(1):54–88, 2004.

[23] J. Nelson and J.R. Hayes. How the writing context shapes college students' strategies for writing from sources. Technical Report 16, Center for Study of Writing. Berkeley,CA: University of California, 1988.

[24] Michael O'Mahony, Neil Hurley, Nicholas Kushmerick, and Gunol Silvestre. Collaborative recommendation: A robustness analysis. *ACM Transactions on Internet Technology*, 4:2004, 2003.

[25] David Pennock, Eric Horvitz, Steve Lawrence, and C Lee Giles. Collaborative filtering by personality diagnosis: A hybrid memory- and model-based approach. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, pages 473–480. Morgan Kaufmann, 2000.

[26] Bradley J. Rhodes and Thad Starner. Remembrance agent: A continuously running automated information retrieval system. In *Proceedings of The First International Conference on The Practical Application Of Intelligent Agents and Multi Agent Technology (PAAM '96)*, pages 487–495, 1996.

[27] J. S. Sachs. Recognition memory for syntactic and semantic aspects of connected discourse. *Perception and Psychophysics*, 2:437–442, 1967.

[28] R. Tagliacozzo. Self-citations in scientific literature. *Journal of Documentation*, 33(4):251–265, 1977.

[29] Simone Teufel. Task-based evaluation of summary quality: Describing relationships between scientific papers. In *Workshop Automatic Summarization, NAACL*, pages 12–21, 2001.

[30] Simone Teufel and Marc Moens. Argumentative classification of extracted sentences as a first step towards flexible abstracting. In *Advances in automatic Text Summarization*, pages 155–171. MIT Press, 1999.

[31] Roberto Torres, Sean M. McNee, Mara Abel, Joseph A. Konstan, and John Riedl. Enhancing digital libraries with techlens+. In *JCDL '04: Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries*, pages 228–236, New York, NY, USA, 2004. ACM.

[32] Michael B. Twidale, Anatoliy A. Gruzd, and David M. Nichols. Writing in the library: Exploring tighter integration of digital library use with the writing process. *Inf. Process. Manage.*, 44(2):558–580, 2008.

[33] Allison Woodruff, Rich Gossweiler, James Pitkow, Ed H. Chi, and Stuart K. Card. Enhancing a digital book with a reading recommender. In *CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 153–160, New York, NY, USA, 2000. ACM.

[34] Xiaolong Zhang, Yan Qu, C. Lee Giles, and Piyou Song. Citesense: supporting sensemaking of research literature. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 677–680, New York, NY, USA, 2008. ACM.

Appendix: Experiment Instructions and Questionnaire

Instructions

How to use ActiveCite:

- The main interface is divided into two parts: editing tool upside and global suggestion window at the bottom. You can click the "refresh" button in the global suggestion window to retrieve a list of suggested papers based on the whole content you have written so far.
- When you think it is necessary to cite a reference at some location in your written content, one method to cite the target paper as you perceived is that check the list in the global suggestion window and if it exists, then just drag and drop the icon in the table to the specific location in your text. The other method is that type [ref] in the specific location as a notation for citation. ActiveCite will highlight it in red color to indicate its

searching status while highlight it in blue color to indicate its completed-searching status. However, you can continue your writing as ActiveCite is performing the search operation in the background. You can click the blue marker to show the local suggestion window. If you find the target paper in the local suggestion list, just check the checkbox to cite it. The initial search terms for the blue marker are underlined while the local suggestion window is open. And you can also refine the search results by filling the form in ”Contextual Search” tab of the local suggestion window and then clicking the ”Search” button.

- For each paper either in the global suggestion list or in local suggestion list, you can click the title column to view the details for that specific paper in another popped up window.

- If you want to delete an already cited paper, just find it in the right marker position and de-check the checkbox corresponding to the paper's row in the local suggestion list. If you want to delete a reference point, click the ”remove” button in the ”Remove the Reference Point” tab of the local suggestion window. All the suggestion information and already cited papers (if any) associated with this point will be deleted.

- You can type [def] at some location as a notation for requiring the definition of the term(s) before this [def] marker. A definition window will pop up when you click the [def] marker and the initial search terms are underlined while the definition window is open. You can refine it by inputting your own terms and click the ”Search” button to refine it in the ”Refine the Search” tab of the definition window.

- You can delete the [def] marker by pressing backspace just as the normal case to delete characters.

- You can click the ”Current Reference” menu item in the ”File” menu located at the top-left of the main interface to view your current reference list.

Experiment task:

1. You can choose one of following two options to use ActiveCite.

    - Option A: Rewrite one paragraph of the related work section in one paper written by you. Try to use ActiveCite to cite the reference(s) in the related work section as many as you can (Maybe some reference cannot be retrieved in the suggestion list).

    - Option B: Write one paragraph which is similar to related work on the site without preparation. Try to use AcitveCite to cite the reference(s) in your paragraph wherever you need.

2. Fill the questionnaire.

Questionnaire

1. Gender: ☐ male ☐ female
2. Age: ☐ 18-24 ☐ 25-34 ☐ 35-44 ☐ 45-60
3. How many papers have you written so far?

   ☐ 1-5 ☐ 5-10 ☐ 11-20 ☐ more than 20
4. Do you find the adding citation time consuming in your real writing paper scenario?

   ☐ 1.very time consuming ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7.not time consuming
5. Do you find it is intuitive to adding citations in ActiveCite?

   ☐ 1.not intuitive ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7.very intuitive
6. Do you find it is useful for ActiveCite to support definition searching function in your paper writing?

   ☐ 1.not useful ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7.very useful
7. Are you satisfied with the accuracy of recommended papers (based on initial underlined terms) which are retrieved from Google Scholar at the current stage in AcitveCite?

   ☐ 1.not satisfied ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7.very satisfied
8. Are you satisfied with the accuracy of recommended papers (based on metadata such as authors, title specified by your refine search terms) which are retrieved from Google Scholar at the current stage in ActiveCite?

   ☐ 1.not satisfied ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7.very satisfied
9. Do you find the system useful to help you manage the citation during your paper writing?

   ☐ 1.not useful ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7.very useful
10. Will you consider using ActiveCite for your paper writing supposed its completely developed version is with improved backend for recommending papers and full editing

features which are similar to traditional editing tools such as Word and LaTex?

☐ 1.no ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7.yes

Comments and Suggestions:

THANK YOU FOR YOUR TIME AND ATTENTION