Sighting Citation Sites
— A Collective-Intelligence Approach for Automatic Summarization of Research Papers using C-Sites —

Dain Kaplan and Takenobu Tokunaga

Department of Computer Science, Tokyo Institute of Technology
{dain, take}@cl.cs.titech.ac.jp

Abstract. This paper presents an architecture for and details of a partially implemented system for automatic research paper summarization harnessing collective intelligence by utilizing the relationship and content between a given paper and citations in other papers that refer to it. We survey pre-existing components related to citation parsing and extraction, introduce our system Cite-Sum, and discuss initial implementation details.

1 Background

In the academic world previous work is almost always the basis for subsequent research. Even in newly emerging fields, prior research is referred to and built upon. This hierarchy of references is stored intrinsically in research papers, and easily extractable to create an interconnected graph via their bibliographies. But in what way and to what extent is each work referenced? What part is predominately the focus of subsequent research, and was it the main theme of a paper, or just a fleeting comment? This information is of course available to those that read a paper, and then search out other papers that cite it, but the process is laborious at best. We believe this information is obtainable simply from analyzing the content of the citations themselves, and their relationships to the original work. Qazvinian and Radev, who are pursing a similar goal through a different approach, also comment that summarization of articles (and subsequently topics) is important for quickly tackling a new field of interest [1].

There has been work in the past [2, 3] on extracting citations and the network they naturally produce. Research has in fact extended to include processing beyond the literal list of citations and their relations. Nanba et al. [4, 5] attempt automatically classifying the collection of all citation areas related to a certain work, where a citation area is defined as the block of text within the research paper that mentions the previous work and is responsible for the appearance of the bibliographical entry. It has also been noted that interdisciplinary links in research are growing [6], and thus the complexity and scale of their interaction. This indicates all the more reason to have a means for better classifying and organizing these relations and for summarizing their content so that a human researcher can grasp the concepts in reasonable time. Past research [4, 5] has

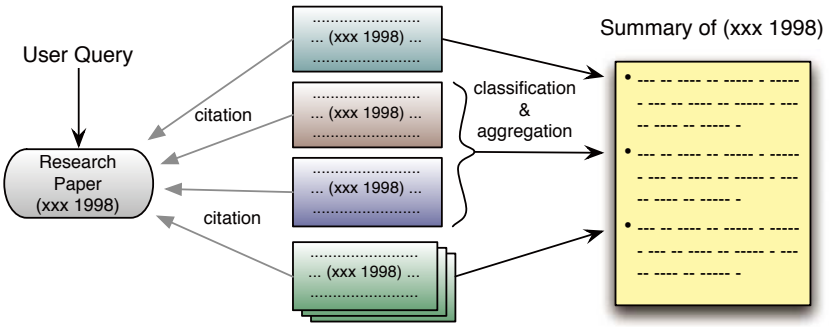

Fig. 1. Diagram of system flow

attempted to classify research papers into groups based on the type of citation, but has used shallow techniques without much emphasis on deeper extraction of summarized content.

2 Overview

We are developing a system, named Cite-Sum, to automatically summarize any and all available citations that refer to a queried paper, which is in the form of a title of a scientific work; the system then attempts to collect all research papers that refer to this paper. The queried paper will be referred to in various contexts [7][1]; the authors of citing papers summarize the work, explain its contributions, and/or its shortcomings, and so forth. In effect, providing a user-based summary of the queried paper. In this manner the larger the database of papers, the better the results. In fact, as research is a continually progressing entity, performing the same search subsequent days may provide new information, as the database grows and more research is performed, improving the results automatically. In other words, "the more users, the better the results" [8].

Thus, an important step of harnessing this "wisdom of crowds" [9] is proper identification of all citations. When previous work is referred to in a research paper, the author must label the content as a citation, and in some fashion, either as footnotes or in a references section provide details to the cited source. We are concerned in this research with the former, which we call "citation sites." Previous research has called these areas simply "citations" or "citing areas," either being too nondescript to explain its nature, or too easily confused with the reference section at the end of a work. We have thus opted for the term "citation site" (or "c-site") indicative of an "excavation site" in which archeologists unearth important clues and information; the boundaries may also be unclear in many cases. We further define the citation string itself, e.g. "(Fakeman et al. 2000)" or "[57]", as the c-site anchor, the place where the c-site originates. These

[1] Weinstock devised a comprehensive list of sixteen citation types

terms allow for further discussion with ambiguity removed. We propose a system that will extract all c-sites referring to a queried paper from other research papers, and further classify and aggregate this extracted content to summarize the paper, as is shown in Figure 1. As this problem concerns aggregating similar content from many nodes in a network to show their relation, this technique has potential for other broader applications for the world wide web in general. Since large online databases of papers are being used, the field and scope of searchable papers should be large, and the recall predictably quite high.

3 System Description

The below process is a step by step explanation of what the system is proposed to do in realtime, and its mechanics, based on a user inputted query.

(1) **Input of query by user**
The system will first receive, as an input from the user, the title of a research paper ("queried paper").

(2) **Retrieval of works with relevant citations**
Given this query, the system will then poll for papers ("citation sources" or "c-sources") having citations referring to the queried paper. For this task, an amply large data set of research papers and indexed citations is necessary. Luckily today there are many online resources to provide this. A benefit of using online resources is that as they improve, the recall and precision of our system will necessarily also improve. This step also includes conversion (from PDF) to text.

(3) **Extraction of c-sites**
The subsequent step is to extract the blocks of text that compose the c-sites from all c-sources. This step is composed of two sub-steps: 1) Finding the c-site by locating the c-site anchor, and 2) Extracting the c-site as it extends from this c-site anchor, both before and after its occurrence. Techniques from extant research for both these sub-steps will be implemented and compared for efficiency, as well as testing our own methods using anaphora chains and/or RST (Rhetorical Structure Theory) analysis for determining the c-site. We also implement a crude, baseline method for comparison that treats the entire paragraph containing the c-site anchor as the c-site.

(4) **Classification of citations**
The c-sites will be analyzed to determine the type of citation [4, 5, 7]. This is useful in determining what effect the work has on the paper citing it. E.g. if it is merely pulling a term, condoning, summarizing, or expanding upon research, etc. In addition, it will pre-parse the c-sites into groups for further analysis in step five.

(5) **Aggregation and Summarization**
In this step, the content of each c-site will be analyzed, and those that evaluate above a certain threshold of similarity will be aggregated for display in the final step. A baseline will be introduced, aggregating two c-sites only if their content matches exactly.

	Google Scholar	CiteSeerX	C.S. Bib
Found	15/16	5/16	11/16
Avg. number of citation	227.4	81.2	N/A
Available for download	$> 60\%$	406/406	N/A

Table 1. Comparison of number of retrieved papers

(6) **Displaying results**
The summarized content from the collected research papers is returned to the user. Collective-intelligence means that the more papers collected, the better should be the results.

4 Existing Tools

One goal of our project is to use existing tools as much as possible. First, we needed a system to crawl websites, follow links, find, and then download papers (PDFs). We decided on Java, and to use the HTMLUnit Framework [10], which allowed us to easily create a solution to do this.

For collecting works that cite the queried paper, we compared three online sources, searching for sixteen seminal works sampled from a publication on statistical language learning [11], and compared the results, as seen in Table 1.

The Collection of Computer Science Bibliographies (C.S. Bib) [12] did not provide a list of citing works, disqualifying it as a suitable candidate. We decided to initially support CiteSeerX [13], even though Google Scholar's [14] estimated paper availability exceeded that of CiteSeerX's (60%[2] of 227.4 equates to 136 c-sources, compared to 81.2). This decision was due to the consistency and ease with which the papers from CiteSeer could be obtained programatically (all 406 c-sources were available), and that after a simple survey of sources labeled as citations in Google Scholar it appeared that not all sources contained a bibliographic entry for the queried paper, which may have introduced further complications.

Next, we needed to extract c-sites. But first, the collected papers (in PDF format) must be converted to text. To do this, we surveyed several tools and selected the two that seemed the most mature, in addition to the pipeline conversion process used by Presri [4, 5] (based on pdftohtml) they were generous enough to donate for our research.[3] Each of these methods has its strengths and weaknesses, however. In addition, there is no easy evaluation metric for directly and objectively comparing them to one another, so we test each with our methods for evaluation of step 3 to find a combined performance metric (see

[2] Google Scholar's paper available of 60% was obtained manually by randomly selecting a paper from the sixteen, and subsequently randomly selecting search result pages, counting the number of links to PS or PDF files and averaging the results.

[3] There also exist online web resources such as PDFTextOnline [15], and Adobe (`http://www.adobe.com/products/acrobat/access_onlinetools.html`) features an online conversion tool as well.

	Baseline						ParsCit						Baseline+ParsCit					
Paper ID	1	2	3	4	5	Total	1	2	3	4	5	Total	1	2	3	4	5	Total
PDFBox	32	16	26	48	3	125	31	0	2	24	0	57	47	16	26	59	3	151
Xpdf	33	15	27	64	3	142	0	0	0	0	0	0	33	15	27	64	3	142
Presri	24	12	22	48	0	106	6	0	2	6	0	14	26	12	23	52	0	113

Table 2. Numbers of extracted c-sites

Table 2). It was also necessary to run the output of PDFBox [16] through a series of regular expressions for adjusting whitespace.

For finding c-sites, Councill et al. created a system called ParsCit [17], and for extracting them, Nanba et al. [4] developed a technique based on matching cue-phrases, which we implemented following its procedural definitions as closely as possible. As a baseline we adopted a simple algorithm for matching author name and publication year for finding c-site anchors using regular expressions, and a crude method of counting the paragraph in its entirety where it occurs, as the c-site. We then ran tests with the above two methods and their combination, with the PDF-to-text-converters, using five papers taken from *Computational Linguistics: Special Issue on the Web as Corpus*, Volume 29, Number 3, 2003 as our data set, chosen for its aptly appropriate title, and compared the results, shown in Table 2.

5 Evaluation Results

The preliminary results show that on average, PDFBox provides the best basis for both ParsCit and the baseline. Xpdf's [18] poor performance was due to its liberal removal of newlines. The Presri-based converter had a similar problem. ParsCit only parses the reference section successfully when it spans many lines, which resulted in lower scores for these two convertors.

Table 2 shows that the best results were achieved by combining the baseline with ParsCit. They were combined in a pipeline fashion, running ParsCit only if the baseline found no c-sites for a c-source. No c-site is counted more than once as a result of combining the two methods. The baseline method is flexible, which allowed it to match more than ParsCit, but also allowed for mistakes. Take the following three citations (matched portions in italics): "*Way and Gough (2003*, 2005a, 2005b)", "Hearne and *Way (2003*, 2006)", and "*Way & Gough, 2003*". ParsCit could match none of these, though the second one was matched in error by the baseline. A common mistake where ParsCit matched only the most literal, and the baseline mis-matched, was the case when two or more papers by the same author, for the same year, with different coauthors are cited. One might be written as "Fakeman and Noman (2008)", while another as "Fakeman et al. (2008)". They may or may not refer to different papers, and only through close analysis of the references section can this difference be discerned.

A cursory look at the accuracy of finding c-sites is shown in Table 3. Though the Xpdf-based approach finds 27 c-sites, two are in error, one resulting from a

correct/matched	Baseline	ParsCit	Baseline+ParsCit	Actual c-sites
PDFBox	25/26	2/2	25/26	
Xpdf	25/27	0	25/27	27
Presri	21/22	1/2	21/23	

Table 3. Precision of identifying c-sites for paper 3

mangled references section. The presri-based system also mangles the references section, and causes ParsCit to misidentify a citation, accounting for the extra mismatch for both the combined and ParsCit approaches. The one citation not found by any of the methods, was "Way & Gough 03", in which the year is specified as only two digits. A more in depth analysis is still needed.

6 Discussion

We have proposed our six step system Cite-Sum for a collective intelligence approach to automatic summarization of research papers, and discussed an implementation of the first three steps using off-the-shelf tools when able. We have analyzed these tools for efficiency and combined them into a base system that can collect c-sites from papers that cite the queried paper from an online resource (CiteSeerX). The system is already capable of showing summaries of the queried paper from many others points of view and in various contexts. Though it has yet to aggregate this data, it shows the potential for harnessing the knowledge and work of others in generating automatic summarization of research papers. We plan to test an implementation of our idea to use anaphora chains and/or RST-based analysis for determining c-site boundaries, at which time we will measure the success of all extraction methods. Extant research on classification of c-sites [4] includes no tools, so we must develop our own. After this we plan to continue with steps five and six. All project information, as well as a planned beta release of the application for general public use, will be available on the project website at: `http://www.cl.cs.titech.ac.jp/cite-sum`.

Acknowledgements

The authors would like to express appreciation to Microsoft for their contribution to this research by selecting it as a recipient of the 2008 WEBSCALE Grant [19].

References

1. Qazvinian, V., Radev, D.R.: Scientific paper summarization using citation summary networks. (2008)
2. Garfield, E., Sher, I.H., Torpie, R.J.: The use of citation data in writing the history of science, Philadelphia, Pennsylvania, Institute for Scientific Information (1964)
3. Small, H.: Co-citation in the scientific literature: A new measure of the relationship between two documents. JASIS **24** (1973) 265–269
4. Nanba, H., Kando, N., Okumura, M.: Classification of research papers using citation links and citation types: Towards automatic review article generation. In: Proceedings of 11th SIG/CR Workshop. (2000) 117–134
5. Nanba, H., Abekawa, T., Okumura, M., Saito, S.: Bilingual presri integration of multiple research paper databases. In: Proceedings of RIAO 2004, Avignon, France (2004) 195–211
6. White, H.D.: Citation analysis and discourse analysis revisited. Applied Linguistics **25**(1) (2004) 89–116
7. Weinstock, M.: Citation indexes. Encyclopedia of Library and Information Science **5** (1971) 16–41
8. O'Reilly, T.: What is web 2.0 (2005) `http://www.oreillynet.com`.
9. Surowiecki, J.: The wisdom of crowds: Why the many are smarter than the few and how collective wisdom shapes business, economies, societies, and nations. Doubleday, New York (2004)
10. HTMLUnit: `http://htmlunit.sourceforge.net`.
11. Charniak, E.: Statistical language learning. The MIT Press (1993)
12. The Collection of C.S. Bibliographies: `http://liinwww.ira.uka.de/bibliography/`.
13. SiteSeerX: `http://citeseerx.ist.psu.edu`.
14. Google Scholar: `http://scholar.google.com`.
15. PDFTextOnline: `http://pdftextonline.com/q/`.
16. PDFBox: `http://www.pdfbox.org/userguide/text_extraction.html`.
17. Councill, I., Giles, C.L., Kan, M.Y.: Parscit: an open-source crf reference string parsing package. In: Proceedings of LREC 2008. (2008)
18. Xpdf: `http://www.foolabs.com/xpdf/`.
19. Web-Scale NLP 2008: `http://research.microsoft.com/ur/asia/research/NLP.aspx`.