A Citation-based Approach to Automatic Paper Summarization

Dain Kaplan and Takenobu Tokunaga

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

Abstract

We propose a novel, collective-intelligence-based approach for automatic summarization of research papers. Each time a citation appears in a paper it is accompanied by a span of text describing the work being cited. These spans of text we call "c-sites." Given a target paper, our method extracts all c-sites in other papers that refer to the target paper, and aggregates them, forming a kind of summary of the target paper. This summary implicitly contains multiple points-of-view, i.e. critical analyses by other scholars, something not present in traditional paper summarization techniques. We also survey several pre-existing components related to citation parsing and extraction, introduce our system Cite-Sum, and discuss its initial implementation details.

1. Background

Research is not an isolated task. Previous work is referenced, refuted, and built upon. Through this process progress is made. This chain of advancement lies latent within the bibliographies and citations of all this accumulating research. But this idea is far from new (Garfield et al., 1964; Small, 1973). There is in fact a wealth of literature on citation analysis in a variety of fields, with interdisciplinary citations increasing (White, 2004; Garfield, 1979). Papers have been topically clustered by extracting citation networks, such as by bibliographic coupling (Kessler, 1963), and co-citation analysis (Small, 1973; White and Griffith, 1981). Conceptual definitions for citation motivation have been devised (Weinstock, 1971; Hodges, 1972; Garfield, 1979; Small, 1982), and automatic methods for their extraction researched (Nanba et al., 2000; Nanba et al., 2004; Teufel et al., 2006), which often involve the identification of cue-phrases. The list of ideas and those that researched them goes on and on.

It is generally agreed that brute citation counting is a poor measure of a paper's importance in a field, which led to research in determining citation motivation. But what about using citations for more than determining a paper's importance? What about using the *content* of citations to summarize the work itself? As the amount of research grows, and the interdisciplinary links multiply, it seems increasingly important to have a means for managing all this information. 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 (Qazvinian and Radev, 2008). We believe that by looking at the content of citations for a given work, beyond their motivation, we can glimpse its importance *and* its important points.

2. Overview

We have devised a process to automatically summarize a given research paper ("queried paper") by using nothing but the collection of citations that refer to it. The running text of the queried paper is never used in the creation of the summary, though it is the target of the summarization task. We believe this has several benefits over traditional summarization techniques, and over using merely the queried paper's abstract as an overall summary of the work, which is written by the author, and lacks objective analysis/critique by definition. For one, the queried paper will be referred to within the collected citations in various contexts (Weinstock, 1971) [1] and in various ways; the authors of citing papers summarize the work, explain its contributions, and/or its shortcomings, and so forth. These citations approach the work from multiple points-of-view, i.e. create a peer-based summary of the queried paper. In this manner the larger the database of research papers (or "peers" as it were) the better the results. Since research is continually advancing, performing the same search subsequent days may provide new information, improving the results automatically. In other words, "the more users, the better the results" (O'Reilly, 2005), where "users" here of course refers to academics. This approach also rings true with the trend of "web as corpus" (Kilgarriff and Grenfenstette, 2003) popular in today's NLP, as well as the notion of "collective-intelligence" epitomized by sites like Wikipedia (Wikipedia, 2001).

2.1. Citation-Sites

Thus, an important step of harnessing this "wisdom of crowds" (Surowiecki, 2004) is proper identification of 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 content labeled as a citation, which we call "citation sites." Previous research has called these areas simply "citations" or "citing areas," either being too nondescript to explain its nature, potentially spanning many sentences, 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 "c-site anchor", the place where the c-site originates. These terms allow for further discussion with ambiguity removed.

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

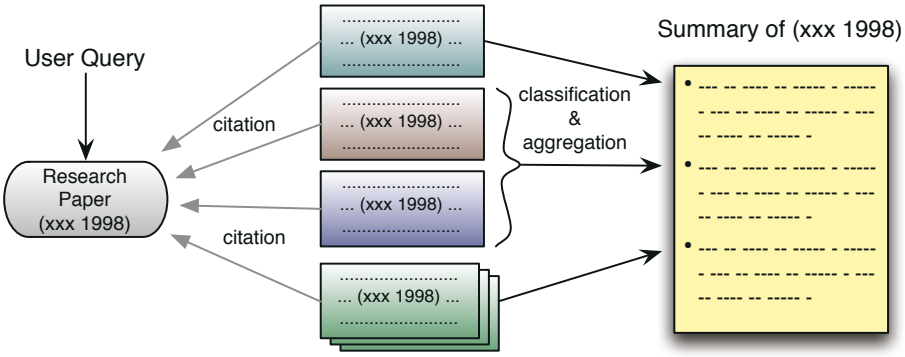

Figure 1: Diagram of system flow

2.2. Cite-Sum

We are developing a system – named Cite-Sum – 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, this technique has potential for other applications related to the world wide web. 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 fetch 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 involve using cue-phrases for demarcating the c-site. A crude baseline for comparison that treats the entire paragraph containing the c-site anchor as the c-site is also needed.

(4) **Classification of citations**
The c-sites will be analyzed to determine the type of citation (its motivation) (Weinstock, 1971; Nanba et al., 2000; Nanba et al., 2004; Teufel et al., 2006). This is useful in determining what effect the work has on the paper citing it. E.g. if it is merely pulling a term, criticizing, summarizing, or expanding upon research, etc. Cue-phrases for this step are also used, and based on existing research. 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 aggregating two c-sites only if their content matches exactly is necessary for benchmarking.

(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. Survey of 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 (HTMLUnit, 2002), 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 (Charniak, 1993), and compared the results, as seen in Table 1.

	Google Scholar	CiteSeerX	C.S. Bib
Found	15/16	5/16	11/16
Citation Avg.	227.4	81.2	N/A
Downloadable	$> 60\%$	406/406	N/A

Table 1: Comparison of number of retrieved papers

The Collection of Computer Science Bibliographies (C.S. Bib) (The Collection of C.S. Bibliographies, 1995) did not

provide a list of citing works, disqualifying it as a suitable candidate. We decided to initially support CiteSeerX (SiteSeerX, 2007), even though Google Scholar's (Google Scholar, 2004) 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 (Nanba et al., 2000; Nanba et al., 2004) (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 Table 2). It was also necessary to run the output of PDFBox (PDFBox, 2002) through a series of regular expressions for adjusting whitespace.

For finding c-sites, Councill et al. created a system called ParsCit (Councill et al., 2008), and for extracting them, Nanba et al. (Nanba et al., 2000) 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 (Xpdf, 1996) 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 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 (Nanba et al., 2000) 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 (Web-Scale NLP, 2008).

[2]Google Scholar's paper availability 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 (PDFTextOnline, 2006), 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

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

7. References

Eugene Charniak. 1993. *Statistical language learning*. The MIT Press.

Isaac Councill, C. Lee Giles, and Min-Yen Kan. 2008. Parscit: an open-source crf reference string parsing package. In *Proceedings of LREC 2008*.

Eugene Garfield, Irving H. Sher, and Richard J. Torpie. 1964. The use of citation data in writing the history of science. Technical report, Institute for Scientific Information, Philadelphia, Pennsylvania.

Eugene. Garfield. 1979. *Citation Indexing: Its Theory and Application in Science, Technology, and Humanities*. Wiley&Sons, Inc.

Google Scholar. 2004. `http://scholar.google.com`.

T. L. Hodges. 1972. *Citation Indexing: Its Potential for Bibliographical Control*. Ph.D. thesis, University of California at Berkeley.

HTMLUnit. 2002. `http://htmlunit.sourceforge.net`.

M. M. Kessler. 1963. Bibliographic coupling between scientific papers. *American Documentation*, 14(1):10–25.

Adam Kilgarriff and Gregory Grenfenstette. 2003. Introduction to the special issue on the Web as corpus. *Computational Linguistics*, 29(3):333–347.

Hidetsugu Nanba, Noriko Kando, and Manabu Okumura. 2000. Classification of research papers using citation links and citation types: Towards automatic review article generation. In *Proceedings of 11th SIG/CR Workshop*, pages 117–134.

Hidetsugu Nanba, Takeshi Abekawa, Manabu Okumura, and Suguru Saito. 2004. Bilingual presri integration of multiple research paper databases. In *Proceedings of RIAO 2004*, pages 195–211, Avignon, France.

Tim O'Reilly. 2005. What is web 2.0. `http://www.oreillynet.com`.

PDFBox. 2002. `http://www.pdfbox.org/userguide/text_extraction.html`.

PDFTextOnline. 2006. `http://pdftextonline.com/q/`.

Vahed Qazvinian and Dragomir R. Radev. 2008. Scientific paper summarization using citation summary networks. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 689–696.

SiteSeerX. 2007. `http://citeseerx.ist.psu.edu`.

Henrry Small. 1973. Co-citation in the scientific literature: A new measure of the relationship between two documents. *Journal of the American Society for Information Science*, 24(4):265–269.

H. Small. 1982. Citation context analysis. *Progress in Communication Science*, 3:287–310.

James Surowiecki. 2004. *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.

Simone Teufel, Advaith Siddharthan, and Dan Tidhar. 2006. Automatic classification of citation function. In *In Proceedings of EMNLP-06*.

The Collection of C.S. Bibliographies. 1995. `http://liinwww.ira.uka.de/bibliography/`.

Web-Scale NLP. 2008. `http://research.microsoft.com/ur/asia/research/NLP.aspx`.

M. Weinstock. 1971. Citation indexes. *Encyclopedia of Library and Information Science*, 5:16–41.

H. D. White and B. C. Griffith. 1981. Author cocitation: A literature measure of intellectual structure. *Journal of the American Society for Information Science*, 32:163–171.

Howard D. White. 2004. Citation analysis and discourse analysis revisited. *Applied Linguistics*, 25(1):89–116.

Wikipedia. 2001. `http://www.wikipedia.org`.

Xpdf. 1996. `http://www.foolabs.com/xpdf/`.