Automatic Extraction of Citation Contexts for Research Paper Summarization: A Coreference-chain based Approach

Dain Kaplan **Ryu Iida** **Takenobu Tokunaga**
Department of Computer Science
Tokyo Institute of Technology
{dain,ryu-i,take}@cl.cs.titech.ac.jp

Abstract

This paper proposes a new method based on coreference-chains for extracting citations from research papers. To evaluate our method we created a corpus of citations comprised of citing papers for 4 cited papers. We analyze some phenomena of citations that are present in our corpus, and then evaluate our method against a cue-phrase-based technique. Our method demonstrates higher precision by 7–10%.

1 Introduction

Review and comprehension of existing research is fundamental to the ongoing process of conducting research; however, the ever increasing volume of research papers makes accomplishing this task increasingly more difficult. To mitigate this problem of information overload, a form of knowledge reduction may be necessary.

Past research (Garfield et al., 1964; Small, 1973) has shown that citations contain a plethora of latent information available and that much can be gained by exploiting it. Indeed, there is a wealth of literature on topic-clustering, e.g. bibliographic coupling (Kessler, 1963), or co-citation analysis (Small, 1973). Subsequent research demonstrated that citations could be clustered on their quality, using keywords that appeared in the running-text of the citation (Weinstock, 1971; Nanba et al., 2000; Nanba et al., 2004; Teufel et al., 2006).

Similarly, other work has shown the utility in the IR domain of ranking the relevance of cited papers by using supplementary index terms extracted from the content of citations in citing papers, including methods that search through a fixed character-length window (O'Connor, 1982; Bradshaw, 2003), or that focus solely on the sentence containing the citation (Ritchie et al., 2008) for acquiring these terms. A prior case study (Ritchie et al., 2006) pointed out the challenges in proper identification of the full span of a citation in running text and acknowledged that fixed-width windows have their limits. In contrast to this, endeavors have been made to extract the entire span of a citation by using cue-phrases collected and deemed salient by statistical merit (Nanba et al., 2000; Nanba et al., 2004). This has met in evaluations with some success.

The Cite-Sum system (Kaplan and Tokunaga, 2008) also aims at knowledge reduction through use of citations. It receives a paper title as a query and attempts to generate a summary of the paper by finding citing papers[1] and extracting citations in the running-text that refer to the paper. Before outputting a summary, it also classifies extracted citation text, and removes citations with redundant content. Another similar study (Qazvinian and Radev, 2008) aims at using the content of citations within citing papers to generate summaries of fields of research.

It is clear that merit exists behind extraction of citations in running text. This paper proposes a new method for performing this task based on coreference-chains. To evaluate our method we created a corpus of citations comprised of citing papers for 4 cited papers. We also analyze some phenomena of citations that are present in our corpus.

The paper organization is as follows. We first define terminology, discuss the construction of our corpus and the results found through its analysis, and then move on to our proposed method using coreference-chains. We evaluate the proposed method by using the constructed corpus, and then conclude the paper.

[1] Papers are downloaded automatically from the web.

Proceedings of the 2009 Workshop on Text and Citation Analysis for Scholarly Digital Libraries, ACL-IJCNLP 2009, pages 88–95,
Suntec, Singapore, 7 August 2009.

2 Terminology

So that we may dispense with convoluted explanations for the rest of this paper, we introduce several terms.

An *anchor* is the string of characters that marks the occurrence of a citation in the running-text of a paper, such as "(Fakeman 2007)" or "[57]".[2] The sentence that this anchor resides within is then the *anchor sentence*. The citation continues from before and after this anchor as long as the text continues to refer to the cited work; this block of text may span more than a single sentence. We introduce the *citation-site*, or *c-site* for short, to represent this block of text that discusses the cited work. Since more than once sentence may discuss the cited work, each of these sentences is called a *c-site sentence*. For clarity will also call the anchor the *c-site anchor* henceforth. A *citing paper* contains the c-site that refers to the *cited paper*. Finally, the *reference* at the end of the paper provides details about a c-site anchor (and the c-site).

Figure 1 shows a sample c-site with the c-site anchor wavy-underlined, and the c-site itself italicized; the non-italicized text is unrelated to the c-site. The reference for this c-site is also provided below the dotted line. In all subsequent examples, the c-site will be in italics and the current place of emphasis wavy-underlined.

> "...Our area of interest is plant growth. *In past research* (Fakeman et al., 2001), *the relationship between sunlight and plant growth was shown to directly correlate. It was also shown to adhere to simple equations for deducing this relationship, the equation varying by plant.* We propose a method that ..."
>
> ....
>
> J. Fakeman: Changing Plant Growth Factors during Global Warming. In: *Proceedings of SCANLP 2001.*

Figure 1: A sample c-site and its reference

3 Corpus Construction and Analysis

We created a corpus comprised of 38 papers citing 4 (cited) papers taken from *Computational Linguistics: Special Issue on the Web as Corpus*, Volume 29, Number 3, 2003 as our data set and preprocessed it to automatically mark c-site anchors to facilitate the annotation process. The citing papers were downloaded from CiteSeer-X;[3] see Table 1 for details.

We then proceeded to manually annotate the corpus using SLAT (Noguchi et al., 2008), a browser-based multi-purpose annotation tool. We devised the following guidelines for annotation. Since the tool allows for two types of annotation, namely *segments* that demarcate a region of text, and *links*, that allow an annotator to assign relationships between them, we created four segment types and three link types. Segments were used to mark c-site anchors, c-sites, background information (explained presently), and references. We used the term *background information* to refer to any running-text that elaborates on a c-site but is not strictly part of the c-site itself (refer to Figure 2 for an example). Even during annotation, however, we encountered situations that felt ambiguous, making this a rather contentious issue.

Our corpus had a limited number of background information annotations, or we would likely have experienced more issues. That being said, it is at least important to recognize that such kinds of supplementary content exist (that may not be part of the c-site but is still beneficial to be included), and needs to be considered more in the future.

We then linked each c-site to its anchor, each anchor to its reference, and any background information to the c-site supplemented. We also decided on annotating entire sentences, even if only part of a sentence referred to the cited paper. Table 1 outlines our corpus.

Table 1: Corpus composition

Paper ID	1	2	3	4	Total
Citing papers	2	14	15	7	38
C-sites	3	17	18	12	50
C-site sentences	6	27	33	28	94

To our knowledge, this is the first corpus constructed in the context of paper summarization related to collections of citing papers.[4]

Analysis of the corpus provided some interesting insights, though a larger corpus is required to confirm the frequency and validity of such phenomena. The more salient discoveries are itemized below. These phenomena may also co-occur.

[2] In practice the anchor does not include brackets, though the brackets do signal the start/end of the anchor. This is because multiple anchors may be present at once, e.g. (Fakeman 2007; Noman 2008).

[3] `http://citeseerx.ist.psu.edu`

[4] Though not specific to the task of summarization through use of c-sites, citation corpora have been constructed in the past, e.g. (Teufel et al., 2006).

Background Information Though not strictly part of a c-site, background information may need to be included for the citation to be comprehensible. Take Figure 2 for example (background information is wavy-underlined) for the c-site anchor "(Resnik & Smith 2003)". The authors insert their own research into the c-site (illustrated with wavy-underlines); this information is important for understanding the following c-site sentence, but is not strictly discussing the cited paper. Background information is thus a form of "meta-information" about the c-site.

In well written papers, often the flow of content is gradual, which can make distinguishing background information difficult.

> "...*Resnik and his colleagues (Resnik & Smith 2003) proposed a new approach, STRAND, ...The databases for parallel texts in several languages with download tools are available from the STRAND webpage. Recently they also applied the same technique for collecting a set of links to monolingual pages identified as Russian by http://www.archive.org, and Internet archiving service.* We have evaluated the Russian database produced by this method and identified a number of serious problems with it. *First, it does not identify the time when the page was downloaded and stored in the Internet archive* ..."

Figure 2: A non-contiguous c-site w/ background information (from (Sharoff, 2006))

Contiguity C-sites are not necessarily contiguous. We found in fact that authors tend to insert opinions or comments related to their own work with sentences/clauses in between actual c-site sentences/clauses, that would be best omitted from the c-site. In Figure 2 the wavy-underlined text shows the author's opinion portion. This creates problems for cue-phrase based techniques, as though they detect the sentence *following* it, they fail on the opinion sentence. Incorporation of a leniency for a gap in such techniques may be possible, but seems more problematic and likely to misidentify c-site sentences altogether.

Related/Itemization Authors often list several works (namely, insert several c-site anchors) in the same sentence using connectives. The works may likely be related, and though this may be useful information for certain tasks, it is important to differentiate which material is related to the c-site, and which is the c-site itself.

In Figure 3 the second sentence discusses both c-site anchors (and should be included in both their c-sites); the first sentence, however, contains two main clauses connected with a connective, each clause a different c-site (one with the anchor "[3]" and one with "[4]"). Sub-clausal analysis is necessary for resolving issues such as these. For our current task, however, we annotated only sentences, and so in this example the second c-site anchor is included in the first.

> "... *STRAND system [4] searches the web for parallel text* and [3] extracts translations pairs among anchor texts pointing together to the same webpage. *However they all suffered from the lack of such bilingual resources available on the web* ..."

Figure 3: Itemized c-sites partially overlapping (from (Zhang et al., 2005))

Nesting C-sites may be nested. In Figure 4 the nested citation ("[Lafferty and Zhai 2001, Lavrenko and Croft 2001]") *should* be included in the parent one ("[Kraaij et al. 2002]"). The wavy-underlined portion shows the sentence needed for full comprehension of the c-site.

> "... *In recent years, the use of language models in IR has been a great success [Lafferty and Zhai 2001, Lavrenko and Croft 2001]. It is possible to extend the approach to CLIR by integrating a translation model. This is the approach proposed in [Kraaij et al. 2002]* ..."

Figure 4: Separate c-site anchors does not mean separate c-sites (from (Nie, 2002))

Aliases Figure 5 demonstrates another issue: aliasing. The author redefines how they cite the paper, in this case using the acronym "K&L".

> "... *To address the data-sparsity issue, we employed the technique used in Keller and Lapata (2003, K&L) to get a more robust approximation of predicate-argument counts. K&L use this technique to obtain frequencies for predicate-argument bigrams that were unseen in a given corpus, showing that the massive size of the web outweighs the noisy and unbalanced nature of searches performed on it to produce statistics that correlate well with corpus data* ..."

Figure 5: C-Site with Aliasing for anchor "Keller and Lapata (2003, K&L)" (from (Kehler, 2004))

4 Coreference Chain-based Extraction

Some of the issues found in our corpus, namely identification of background information, non-contiguous c-sites, and aliases, show promise of

Table 2: Evaluation results for coreference resolution against the MUC-7 formal corpus.

	MUC-7 Task			Sentence Eval.		
System Setting	R	P	F	R	P	F
All Features	35.71	74.71	48.33	36.27	80.49	50.00
w/o SOON_STR_MATCH	48.35	83.81	61.32	48.35	88.00	62.41
w/o COSINE_SIMILARITY	46.70	82.52	59.65	46.70	86.73	60.71

resolution with coreference-chains. This is because coreference-chains match noun phrases that appear with other noun phrases to which they refer, a characteristic present in these three categories. On the other hand, cue-phrases do not detect any c-site sentence that does not use keywords (e.g. "In addition"). In the following section we discuss our implementation of a coreference chain-based extraction technique, and how we then applied it to the c-site extraction task. An analysis of the results then follows.

4.1 Training the Coreference Resolver

To create and train our coreference resolver, we used a combination of techniques as outlined originally by (Soon et al., 2001) and subsequently extended by (Ng and Cardie, 2002). Mimicking their approaches, we used the corpora provided for the MUC-7 coreference resolution task (LDC2001T02, 2001), which includes sets of newspaper articles, annotated with coreference relations, for both training and testing. They also outlined a list of features to extract for training the resolver to recognize the coreference relations. Specifically, (Soon et al., 2001) established a list of 12 features that compare a given anaphor with a candidate antecedent, e.g. gender agreement, number agreement, both being pronouns, both part of the same semantic class (i.e. WordNet synset hyponyms/hypernyms), etc.

For training the resolver, a corpus annotated with anaphors and their antecedents is processed, and pairs of anaphor and candidate antecedents are created so as to have only one positive instance per anaphor (the annotated antecedent). Negative examples are created by taking all occurrences of noun phrases that occur *between* the anaphor and its antecedent in the text. The antecedent in these steps is also always considered to be to the left of, or preceding, the anaphor; cataphors are not addressed in this technique.

We implemented, at least minimally, all 12 of these features, with a few additions of what (Ng and Cardie, 2002) hand selected as being most salient for increased performance. We also extended this list by adding a cosine-similarity metric between two noun phrases; it uses bag-of-words to create a vector for each noun phrase (where each word is a term in the vector) to compute their similarity. The intuition behind this is that noun phrases with more similar surface forms should be more likely to corefer.

We further optimized string recognition and plurality detection for handling citation-strings. See Table 3 for the full list of our features. While both (Soon et al., 2001) and (Ng and Cardie, 2002) induced decision trees (C5 and C4.5, respectively) we opted for using an SVM-based approach instead (Vapnik, 1998; Joachims, 1999). SVMs are known for being reliable and having good performance.

4.2 Evaluating the Coreference Resolver

We ran our trained SVM classifier against the MUC-7 formal evaluation corpus; the results are shown in Table 2.

The results using all features listed in Table 3 are inferior to those set forth by (Soon et al., 2001; Ng and Cardie, 2002); likely this is due to poorer selection of features. Upon analysis, it seems that half of the misidentified antecedents were still chosen within the correct sentence and more than 10% identified the proper antecedent, but selected the entire noun phrase (when that antecedent was marked as, for example, only its head); the majority of these cases involved the antecedent being only one sentence away from the anaphor. Since the former seemed suspect of a partial string matching feature, we decided to re-run the tests first excluding our implementation of the `SOON_STR_MATCH` feature, and then our `COSINE_SIMILARITY` feature. The results for this are shown in Table 2. It can be seen that using either of the two string comparison features works substantially better than with both of them in tandem, with the `COSINE_SIMILARITY` feature showing signs of overall better performance which is competitive to (Soon et al.,

Table 3: Features used for coreference resolution.

Feature	Possible Values	Brief Description (where necessary)
ANAPHOR_IS_PRONOUN	T/F	
ANAPHOR_IS_INDEFINITE	T/F	
ANAPHOR_IS_DEMONSTRATIVE	T/F	
ANTECEDENT_IS_PRONOUN	T/F	
ANTECEDENT_IS_EMBEDDED	T/F	Boolean indicating if the candidate antecedent is within another NP.
SOON_STR_MATCH	T/F	As per (Soon et al., 2001). Articles and demonstrative pronouns removed before comparing NPs. If any part of the NP matches between candidate and anaphor set to true (T); false otherwise.
ALIAS_MATCH	T/F	Creates abbreviations for organizations and proper names in an attempt to find an alias.
BOTH_PROPER_NAMES	T/F	
BOTH_PRONOUNS	T/F/–	
NUMBER_AGREEMENT	T/F/–	Basic morphological rules applied to the words to see if they are plural.
COSINE_SIMILARITY	NUM	A cosine similarity score between zero and one is applied to the head words of each NP.
GENDER_AGREEMENT	T/F/–	If the semantic class is Male or Female, use that gender, otherwise if a salutation is present, or lastly set to Unknown.
SEMANTIC_CLASS_AGREEMENT	T/F/–	Followed (Soon et al., 2001) specifications for using basic WordNet synsets, specifically: Female and Male belonging to Person, Organization, Location, Date, Time, Money, Percent belonging to Object. Any other semantic classes mapped to Unknown.

2001; Ng and Cardie, 2002). We exclude the `SOON_STR_MATCH` feature in the following experiments.

However, the MUC-7 task measures the ability to identity the proper antecedent from a list of candidates; the c-site extraction task is less ambitious in that it must only identify if a sentence contains the antecedent, not which noun phrase it is. When we evaluate our resolver using these loosened conditions it is expected that it will perform better.

To accomplish this we reevaluate the results from the resolver in a sentence-wise manner; we group the test instances by anaphor, and then by sentence. If any noun phrase within the sentence is marked as positive when there is in fact a positive noun phrase in the sentence, the sentence is marked as correct, and incorrect otherwise. The results in Table 2 for this simplified task show an increase in recall, and subsequently F-measure. The numbers for the loosened constraints evaluation are counted by sentence; the original is counted by noun phrase only.

Our system also generates many fewer training instances than the previous research, which we attribute to a more stringent noun phrase extraction procedure, but have not investigated thoroughly yet.

4.3 Application to the c-site extraction task

As outlined above, we used the resolver with the loosened constraints, namely evaluating the sentence a potential antecedent is in as likely or not, and not which noun phrase within the sentence is the actual antecedent. Using this principle as a base, we devised an algorithm for scanning sentences around a c-site anchor sentence to determine their likelihood of being part of the c-site. The algorithm, shown in simplified form in Figure 6, is described below.

Starting at the beginning of a c-site anchor sentence `AS`, scan left-to-right; for every noun phrase encountered within `AS`, begin a right-to-left sentence-by-sentence search; prepend any sentence `S` containing an antecedent above a certain likelihood `THRESHOLD`, until `DISTANCE` sentences have been scanned and no suitable candidate sentences have been found. We set the likelihood score to 1.0, tested ad-hoc for best results, and the distance-threshold to 5 sentences, having noted in our corpus that no citation is discontinuous by more than 4.

In a similar fashion, the algorithm then proceeds to scan text following `AS`; for every noun phrase `NP` encountered (moving left-to-right), begin a right-to-left search for a suitable antecedent. If a sentence is not evaluated above `THRESHOLD`,

Table 4: Evaluation results for c-site extraction w/o background information

	Sentence (Micro-average)			C-site (Macro-average)		
Method	R	P	F	R	P	F
Baseline 1 (anchor sentence)	53.2	100	69.4	74.6	100	85.5
Baseline 2 (random)	75.5	58.2	65.7	87.4	71.2	78.5
Cue-phrases (CP)	64.9	64.9	64.9	84.0	80.9	82.4
Coref-chains (CC))	64.9	74.4	69.3	81.0	87.2	84.0
CP/CC Union	74.5	58.8	65.7	88.4	75.0	81.1
CP/CC Intersection	55.3	91.2	69.0	76.6	95.7	85.1

```
set CSITE to AS

pre:
foreach NP in AS
  foreach sentence S preceding AS
    if DISTANCE > MAX-DIST goto post
    if likelihood > THRESHOLD then
      set CSITE to S + CSITE
      reset DISTANCE
    end
  end
end

post:
foreach sentence S after AS
  foreach NP in S
    foreach sentence S2 until S
      if DISTANCE > MAX-DIST stop
      if S2 has link then
        if likelihood > THRESHOLD then
          set S2 has link
        end
      end
    end
  end
end
```

Figure 6: Simplified c-site extraction algorithm using coreference-chains

it will be ignored when the algorithm backtracks to look for candidate noun phrases for a subsequent sentence, thus preserving the coreference-chain and preventing additional spurious chains. If more than DISTANCE sentences are scanned without finding a c-site sentence, the process is aborted and the collection of sentences returned.

4.4 Experiment Setup

To evaluate our coreference-chain extraction method we compare it with a cue-phrases technique (Nanba et al., 2004) and two baselines. Baseline 1 extracts only the c-site anchor sentence as the c-site; baseline 2 includes sentences before/after the c-site anchor sentence as part of the c-site with a 50/50 probability — it tosses a coin for each consecutive sentence to decide its inclusion. We also created two hybrid methods that combine the results of the cue-phrases and coreference-chain techniques, one the union of their results (includes the extracted sentences of both methods), and the other the intersection (includes sentences only for which both methods agree), to measure their mutual compatibility.

The annotated corpus provided the locations of c-site anchors for the cited paper within the citing paper's running-text. We then compared the extracted c-sites of each method to the c-sites of the annotated corpus.

4.5 Evaluation

The results of our experiments are presented in Table 4. We evaluated each method as follows. Recall and precision were measured for a c-site based on the number of extracted sentences; if an extracted sentence was annotated as part of the c-site, it counted as correct, and if an extracted sentence was not part of a c-site, incorrect; sentences annotated as being part of the c-site not extracted by the method counted as part of the total sentences for that c-site. As an example, if an annotated c-site has 3 sentences (including the c-site anchor sentence), and the evaluated method extracted 2 of these and 1 incorrect sentence, then the recall for this c-site using this method would be $2/3$, and the precision $2/(2+1)$.

Since the evaluation is inherently sentence-based, we provide two averages in Table 4. The micro-average is for sentences across all c-sites; in other words, we tallied the correct and incorrect sentence count for the whole corpus and then divided by the total number of sentences (94). This average provides a clearer picture on the efficacy of each method than does the macro-average. The macro-average was computed per c-site (as explained above) and then averaged over the total number of c-sites in the corpus (50).

With the exception of a 3% lead in macro-average recall, coreference-chains outperform cue-phrases in every way. We can see a substantial difference in micro-average precision (74.4 vs. 64.9), which results in nearly a 5% higher F-measure. The macro-average precision is also higher by more than 6%. It matches more and misses far less. The loss in the macro-average recall can be attributed to the coreference-chain method missing one of two sentences for several c-sites, which would lower its overall recall score; keep in mind that since in the macro-average all c-sites are treated equally, even large c-sites in which the coreference-chain method performs well, such an advantage will be reduced with averaging and is therefore misleading.

Baseline 2 performed as expected, i.e. higher than baseline 1 for recall. Looking only at F-measures for evaluating performance in this case is misleading. This is particularly the case because precision is more important than recall — we want accuracy. Coreference-chains achieved a precision of over 87.2 compared to the 71.2 of baseline 2.

The combined methods also showed promise. In particular, the intersection method had very high precision (91.2 and 95.7), and marginally managed to extract more sentences than baseline 1. The union method has more conservative scores.

We also understood from our corpus that only about half of c-sites were represented by c-site anchor sentences. The largest c-site in the corpus was 6 sentences, and the average 1.8. This means using the c-site anchor sentence alone excludes on average about half of the valuable data.

These results are promising, but a larger corpus is necessary to validate the results presented here.

5 Conclusions and Future Work

The results demonstrate that a coreference-chain-based approach may be useful to the c-site extraction task. We can also see that there is still much work to be done. The scores for the hybrid methods also indicate potential for a method that more tightly couples these two tasks, such as Rhetorical Structure Theory (RST) (Thompson and Mann, 1987; Marcu, 2000). Though it has demonstrated superior performance, coreference resolution is not a light-weight task; this makes real-time application more difficult than with cue-phrase-based approaches.

Our plans for future work include the construction of a larger corpus of c-sites, investigation of other features for improving our coreference resolver, and applying RST to c-site extraction.

Acknowledgments

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, 2008).

References

Shannon Bradshaw. 2003. Reference directed indexing: Redeeming relevance for subject search in citation indexes. In *Proceedings of the 7th ECDL*, pages 499–510.

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

Thorsten Joachims. 1999. Making large-scale support vector machine learning practical. In Bernhard Schölkopf, Christopher J. C. Burges, and Alexander J. Smola, editors, *Advances in kernel methods: support vector learning*, pages 169–184. MIT Press, Cambridge, MA, USA.

Dain Kaplan and Takenobu Tokunaga. 2008. Sighting citation sites: A collective-intelligence approach for automatic summarization of research papers using c-sites. In *ASWC 2008 Workshops Proceedings*.

Andrew Kehler. 2004. The (non)utility of predicate-argument frequencies for pronoun interpretation. In *In: Proceedings of 2004 North American chapter of the Association for Computational Linguistics annual meeting*, pages 289–296.

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

LDC2001T02. 2001. Message understanding conference (MUC) 7.

Daniel Marcu. 2000. The rhetorical parsing of unrestricted texts: A surface-based approach. *Computational Linguistics*, 26(3):395–448.

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.

Vincent Ng and Claire Cardie. 2002. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 104–111.

J. Nie. 2002. Towards a unified approach to clir and multilingual ir. In *In: Workshop on Cross Language Information Retrieval: A Research Roadmap in the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 8–14.

Masaki Noguchi, Kenta Miyoshi, Takenobu Tokunaga, Ryu Iida, Mamoru Komachi, and Kentaro Inui. 2008. Multiple purpose annotation using SLAT — Segment and link-based annotation tool —. In *Proceedings of 2nd Linguistic Annotation Workshop*, pages 61–64, May.

John O'Connor. 1982. Citing statements: Computer recognition and use to improve retrieval. *Information Processing & Management.*, 18(3):125–131.

Vahed Qazvinian and Dragomir R. Radev. 2008. Scientific paper summarization using citation summary networks.

Anna Ritchie, Simone Teufel, and Stephen Robertson. 2006. How to find better index terms through citations. In *Proceedings of the Workshop on How Can Computational Linguistics Improve Information Retrieval?*, pages 25–32, Sydney, Australia, July. Association for Computational Linguistics.

Anna Ritchie, Stephen Robertson, and Simone Teufel. 2008. Comparing citation contexts for information retrieval. In *CIKM '08: Proceedings of the 17th ACM conference on Information and knowledge management*, pages 213–222, New York, NY, USA. ACM.

Serge Sharoff. 2006. Creating general-purpose corpora using automated search engine queries. In *WaCky! Working papers on the Web as Corpus*. *Gedit*.

H. Small. 1973. Co-citation in the scientific literature: A new measure of the relationship between two documents. *JASIS*, 24:265–269.

Wee Meng Soon, Daniel Chung, Daniel Chung Yong Lim, Yong Lim, and Hwee Tou Ng. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544.

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

Sandra A. Thompson and William C. Mann. 1987. Rhetorical structure theory: A framework for the analysis of texts. *Pragmatics*, 1(1):79–105.

Vladimir N. Vapnik. 1998. *Statistical Learning Theory*. Adaptive and Learning Systems for Signal Processing Communications, and control. John Wiley & Sons.

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

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

Ying Zhang, Fei Huang, and Stephan Vogel. 2005. Mining translations of oov terms from the web through. In *International Conference on Natural Language Processing and Knowledge Engineering (NLP-KE' 03)*, pages 669–670.