ABSTRACT

Title of thesis: CITATION HANDLING:
PROCESSING CITATION TEXTS
IN SCIENTIFIC DOCUMENTS

Michael Alan Whidby
Master of Science, 2012

Thesis directed by: Professor Bonnie Dorr
Dr. David Zajic
Department of Computer Science

Citation sentences (sentences that cite other papers) play a key role in the summarization of scientific articles. However, a citation-based summarization system that depends on generic natural language processing components, such as parsers or sentence compressors, will perform poorly if those components cannot handle citations correctly.

In this thesis, I examine the effect of citation handling on parsing, sentence compression, and multi-document summarization. There are two types of citations that occur in citation sentences: constituent citations and parenthetical citations. I propose an automatic citation classifier based on training data created through Mechanical Turk tasks. I demonstrate that the use of type-specific citation handling as pre-processing improves the performance of a state-of-the-art generic parser, both for quality of the parse trees and running time. Extrinsic evaluations demonstrate that improving the performance of a parser on citation sentences in turn improves

the performance of a sentence compressor, Trimmer Zajic et al. (2007), and a multi-document summarization system, MASCS, according to several summarization measures.

CITATION HANDLING: PROCESSING CITATION TEXTS IN SCIENTIFIC DOCUMENTS

by

Michael Alan Whidby

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2012

Advisory Committee:
Professor Bonnie Dorr, Chair/Advisor
Dr. David Zajic, Co-advisor
Professor Hal Daumé III

Acknowledgments

The successful completion of this thesis was made possible by the invaluable contributions of a number of people.

First and foremost I'd like to thank my advisor, Professor Bonnie Dorr, for giving me the opportunity to work on challenging and impactful projects over the past two years. Her continuous encouragement, support, and organization since day one has kept me on track and focused on my research and thesis.

I would also like to thank my co-advisor, Dr. David Zajic. Without his guidance and valuable insight, this thesis would have been a distant dream. He was always available to meet and talk outside of our normal meeting time, and those sessions led to a great deal of the work in this thesis. In addition, many thanks to Professor Hal Daumé III for agreeing to serve on my thesis committee and providing helpful comments and thoughts on my work, as well as teaching two of the more influential courses of my academic career (Computational Linguistics and Machine Learning). I would also like to thank Dr. Taesun Moon for his help on various aspects of my work, and for providing interesting avenues for future work in our research group.

I'd also like to thank my many friends who reminded me that graduate school should not take up all of your time, and that going out to relax and unwind is essential to your sanity and well-being. I'm also grateful to my dog and roommate, Bell, who stayed up with me on all those late nights and made sure I went outside every day to get my daily dose of Vitamin D.

Finally, I'd like to thank my family for providing me with the means and opportunity to pursue graduate study, and for supporting me every step of the way.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AAN	ACL Anthology Network
CRF	Conditional Random Fields
DP	Dependency Parsing
MASCS	Multiple Alternate Sentence Compression Summarizer
MDS	Multi-document Summarization
QA	Question Answering
SSL	Semi-supervised Learning

Chapter 1

Introduction

It has become increasingly important to support the needs of users who seek to understand a wide range of scientific areas with which they are not currently familiar. For example, it has become common for interdisciplinary review panels to be called upon to review proposals in a wide range of areas, without access to the most up-to-date summaries (or surveys) of the relevant topics. NLP and visualization tools have been developed to accommodate this need (Gove et al., 2011) and steps have been taken to provide summaries for the purpose of survey creation, but citations that occur in the input texts introduce noise that leads to disfluent summarization output.

In this thesis I present the first steps toward improving summarization of scientific documents through parsing of citation sentences (sentences that cite other papers). Prior work (Mohammad et al., 2009) argues that citation sentences play a crucial role in automatic summarization of a topic area, but did not take into account the noise introduced by the citations themselves. As a first step toward improving the fluency of summarization of citation sentences, I apply two different approaches to citation handling and then examine the effects of these approaches on the parse trees produced by the Stanford Parser (Klein and Manning, 2003). If the parser performs poorly, then a summarization system that uses the parser will

also perform poorly. I demonstrate that the quality of parse trees is improved with citation handling.

In addition, the improved parse trees serve as input to Trimmer (Zajic et al., 2007), a sentence compression system, and MASCS (Multiple Alternate Sentence Compression Summarization), a multi-document summarization system. As such, I demonstrate that the improved parsing output has a positive effect on Trimmer's sentence candidates for summarization of scientific articles. These sentence candidates are evaluated with a language model, and the summaries generated from MASCS are evaluated with ROUGE (Lin, 2004) and Pyramid (Nenkova and Passonneau, 2004). In all cases, using citation handling leads to improved performance compared to that of a summarizer that does not support citation handling.

1.1 Motivation

Citations introduce noise that causes errors in constituency parsers and summarization systems. Like formulas and footnotes in scientific text, citations can also cause unpredictable and incorrect behavior from a summarization system. In this section, I examine some of the problems with citations that arise with parsers and summarization systems.

1.1.1 Parser Issues Caused By Citation Texts

Citations introduce noise into constituency parsers that may cause erroneous parse trees. These sorts of errors include mislabelling the citations themselves or

producing an incorrect tree structure. One common error that occurs with the Stanford Parser (Klein and Manning, 2003) deals with misplacing conjunctions when there are multiple citations.

For example, consider the citation sentence and a portion of the resulting parse tree from the Stanford Parser, shown in Figure 1.1. Here, both the "(CC and)" and the second citation should be attached to under the PP that includes the first citation. A correct version of this subtree would be "(PP in (NP (NP CIT-1) (CC and) (NP CIT-2)))," where CIT-1 is the first citation and CIT-2 is the second citation.

Another example of a misplaced conjunction occurs in the parse tree of the the citation sentence shown in Figure 1.2. In this case, the "(CC and/or)" conjunction has been misplaced: it should attach under the VP that dominates "are deterministic." A correct version of this subtree would be "(VP are (ADJ deterministic) (CC and/or) (ADJ linear))."

With the first citation sentence, the citations are syntactically part of the sentence, but the two citations together could be treated like a conjoined noun phrase. In the case of the second citation sentence, the citations are not syntactically part of the sentence, and therefore add nothing in terms of sentence structure. Treating the citations like a conjoined noun phrase in the first case and ignoring the citations in the second case would improve the parse trees generated for the citation sentence. Improved parse trees would allow a ssentence compression system to better apply syntactic rules to the citation sentence when generating sentence compressions.

```
...
(, ,)
(NP (PRP we))
(VP (VBD revisited)
  (NP
    (NP (NP
        (NP (DT the) (NN test) (NNS cases))
        (VP (VBN discussed)
          (PP (IN in))))
      (PRN (-LRB- -LRB-)
        (NP (NNP Carroll)
          (CC et)
          (NNP al))
        (, ,)
        (NP (CD 1999))
        (-RRB- -RRB-)))
    (CC and)
    (SBAR
      (S
        (VP
          (PRN (-LRB- -LRB-)
            (NP (NNP Koller)
              (CC and)
              (NNP Striegnitz))
            (, ,)
            (NP (CD 2002))
            (-RRB- -RRB-))
          (PP (IN by)
            (NP
              (NP (JJ producing) (JJ similar) (NNS sentences))
              (PP (IN in)
                (NP (NNP French)))))))))
(. .)))
```

Figure 1.1: The parse tree for the citation sentence *"To get an estimate of how our realiser compares with existing published results, we revisited the test cases discussed in [Carroll et al, 1999] and [Koller and Striegnitz, 2002] by producing similar sentences in French."* Notice the misplaced "(CC and)" in the parse tree.

```
(ROOT
  (S
    (ADVP (RB Recently))
    (NP (JJ statistical) (JJ dependency) (NN parsing) (NNS techniques))
    (VP (VBP have)
      (VP (VBN been)
        (VP
          (VP (VBN proposed)
            (SBAR
              (WHNP (WDT which))
              (S
                (VP (VBP are)
                  (ADJP (JJ deterministic))))))
          (CC and/or)
          (VP (VBN linear)
            (PRN (-LRB- -LRB-)
              (NP
                (NP (NNP Yamada)
                  (CC and)
                  (NNP Matsumoto))
                (, ,)
                (NP (CD 2003))
                (, ;)
                (NP (NNP Nivre)
                  (CC and)
                  (NNP Scholz))
                (, ,)
                (NP (CD 2004)))
              (-RRB- -RRB-))))))
    (. .)))
```

Figure 1.2: The parse tree for the citation sentence *"Recently statistical dependency parsing techniques have been proposed which are deterministic and/or linear (Yamada and Matsumoto, 2003; Nivre and Scholz, 2004)."* Notice the misplaced "(CC andor)" in the parse tree.

1.1.2 Summarization Issues Caused by Citation Texts

We currently employ a variant of the Trimmer system (Zajic et al., 2007) that applies syntactic rules to sentences to create sentence-compression candidates for summarization. One syntactic rule that the system uses is a conjunction rule that specifically creates a distinct compressed version for each item in the conjunction. Consider an example citing sentence, "The probability model may be either conditional (Duan et al., 2007) or generative (Titov and Henderson, 2007)." The citation "(Titov and Henderson, 2007)" contains a conjunction. Application of the conjunction rule creates three sentence candidates, two of which now contain erroneous citations:

1. "The probability model may be either conditional (Duan et al., 2007) or generative *(Titov and Henderson, 2007).*" (the original conjunction)

2. "The probability model may be either conditional (Duan et al., 2007) or generative *(Titov, 2007).*"

3. "The probability model may be either conditional (Duan et al., 2007) or generative *(Henderson, 2007).*"

Note that in this case, the sentence candidates are no different from the source sentence in terms of actual content, but the application of the conjunction rule has made the original citations incorrect. A means for avoiding the application of the conjunction rule on *"and"* citations is necessary in order to maintain the integrity of the original citation.

1.2 Types of Citations

There are two different types of citations that are used in citation sentences: *constituent* citations and *parenthetical* citations. Constituent citations (CC) take an overt role in the syntactic structure of a sentence; removing a CC from a sentence would make the sentence ungrammatical. They typically occur as noun phrases and may take on the role of agents who did or claimed something. On the other hand, parenthetical citations (PC) are citations that are structurally independent of the sentence; removing them would not have any effect on the grammaticality of the sentence. They are typically used as an instance of some event or situation mentioned in the sentence.

1.3 Hypothesis

The primary hypothesis underlying this thesis is that citation handling will prove to be useful in correcting the erroneous parse trees like the ones presented in Section 1.1.1, and the parser will be able to generate parses faster with citation handling. Citation handling will also improve the sentence candidates that are produced by a modified version of Trimmer. Finally, the summaries that were generated from MASCS using citation handling will be shown to be superior to those generated without citation handling in terms of two standard summarization measures, ROUGE and Pyramid.

1.4 Contributions

To solve the parser and summarization issues associated with unprocessed citations, this thesis introduces an approach, called *citation handling*, to preprocessing citations. Citation handling involves replacing or removing a citation based on the citation's type. Another contribution is a software implementation of citation handling, including a citation classifier that designates citation as either constituent or parenthetical. With citation handling, this thesis shows that better quality parse trees are created by the Stanford Parser, and with a much faster running time. Additionally, this thesis concludes that these improved parse trees significantly improve the quality and performance of two NLP components, a sentence compressor and a summarization system. These benefits can be extended to any NLP component that relies on parse trees, especially scholarly texts containing citations.

1.5 Roadmap

The rest of this thesis is laid out as follows: Chapter 2 presents related work. In Chapter 3, I describe the training and evaluation of a classifier to determine whether a citation is constituent or parenthetical. Chapter 4 details the citation handling process, and follows an example citation sentence as it goes through the different steps in the process. I investigate the application of citation handling and its effects on three generic NLP components, a parser, sentence compressor, and summarization system in Chapter 5. Specific examples of the benefits of citation handling are also presented for each component. Chapter 6 presents evaluations

on all three of the NLP components on standard evaluation measures. Finally, I conclude and present future work in Chapter 7.

Chapter 2

Related Work

A summary of a scientific article can be produced from two different sources: the scientific article itself, and what other researchers have said about the work presented in the scientific article (via citation sentences). An author can describe what they think to be the important contributions of their paper, whereas citation sentences can capture what others in the field determine to be the contributions of the paper, and provide several different perspectives on the same article (Bradshaw, 2003).

Elkiss et al. (2008) conducted several experiments on PubMed Central articles and found that summaries generated using citation sentences contained more information and cohesion (a lexical similarity metric) than summaries generated from abstracts. Similarly, Mohammad et al. (2009) demonstrated the usefulness of citation sentences to produce a multi-document survey of scientific articles in comparison to producing summaries with abstracts and full texts. Qazvinian and Radev (2008) built a similarity network of the citation sentences that cite a target paper, and applied network analysis techniques to determine the sentences that covered as much summarized facts about the paper as possible. Bradshaw (2002) used citation sentences to determine the content of articles and improve the results of a search engine. Mei and Zhai (2008) used what they termed *citation context*, the collection

of windows of sentences surrounding citation sentences, to perform impact-based summarization. While these works focused on the effectiveness of using citation sentences in various forms of single- and multi-document summarization, they did not consider the effect that citations themselves have on the various components of summarization (e.g., the effect on a parser or sentence compressor).

The aim of this thesis is not to determine the utility of citation sentences as in the prior works cited above, but to determine the impact of proper citation handling within the citation sentences for downstream processing. Specifically, I examine the effects of citation handling as it pertains to the quality and performance of parsing, sentence compression, and multi-document summarization.

Nanba et al. (2004) analyzed citation sentences and proposed three groups of citations based on the reason for the citation. For example, these reasons could be to point out problems in a related work, or to show other author's theories and methodologies. Similarly, Teufel et al. (2006) trained a classifier to group citations by their function into four categories. This thesis presents a classifier that categorizes citations into two types; however, the types of citations in this thesis are based on their syntactic properties, and not the reasoning or intent of the citation.

Abu-Jbara and Radev (2011) perform several preprocessing techniques to citation sentences, such as removing sentences that do not describe any aspect of the author's work they are citing. Another technique they apply is the preprocessing of citations similar to that presented in this thesis. In their approach, a citation is either removed entirely (and not re-inserted later) or replaced with a pronoun (he, she, they).

The approach presented in this thesis preprocesses citations differently - if a citation is removed before parsing, it is later re-inserted back into the sentence candidates for summarization. In addition, citations that are not removed are replaced with a filler text rather than a pronoun, and the original citation text is re-inserted into the sentence compression candidates. The approach described in this thesis uses a classifier-based approach to determine whether a citation should be replaced or removed, while Abu-Jbara and Radev use a heuristic-based approach. A comparison of these two approaches to classifying citations is presented in Chapter **??**. Abu-Jbara and Radev investigated the impact of their preprocessing techniques in their evaluation; however, they did not perform evaluations on the effect of preprocessing the citations on their system. In contrast, this thesis presents numerous evaluations to measure the specific impact of preprocessing citations on parsing, sentence compression, and summarization.

In the next chapter, I examine the two different types of citations that occur in citation snetences, and train and evaluate a citation classifier to distinguish between these types of citations.

Chapter 3

Citation Classification: Data Annotation and Classifier Training

In this chapter, I introduce two different types of citations: constituent citations and parenthetical citations. These types of citations vary in how they are used in the sentence, and what impact they have on the syntax of the sentence. Both types of citations will be presented, along with examples of each citation type. I will then present a series of Mechanical Turk tasks for the annotation of citation data, and the training of a classifier on this data, for the purpose of distinguishing between constitutent and parenthetical citations.

3.1 Types of Citations

Constituent citations (CC) take an overt role in the syntactic structure of a sentence; removing a CC from a sentence would make the sentence ungrammatical. They typically occur as noun phrases and may take on the role of agents who did or claimed something.

Some examples of constituent citations include:

- "As pointed out by *(Lee and Wu, 2007; Gimenez and Marquez, 2003)*, the introduction of suffix features can effectively help to guess the unknown words for tagging and chunking."

- "*Lapata (2003)* ordered sentences based on conditional probabilities of sen-

tence pairs."

- "Rank of a sentence is predicted from regression model built on feature vectors of sentences in the training data using support vector machine as explained in *(Schilder and Kondadandi, 2008)*."

Parenthetical citations (PC) are citations that are structurally independent of the sentence; removing them would not have any effect on the grammaticality of the sentence. They are typically used as an instance of some event or situation mentioned in the sentence.

Some examples of parenthetical citations include:

- "Previous studies pointed out that information from wider scope, at the document or cross-document level, could provide non-local information to aid event extraction *(Ji and Grishman 2008, Liao and Grishman 2010a)*."

- "Some previous work *(Peng et al., 2004; Tseng et al., 2005; Low et al., 2005)* illustrated the effectiveness of using characters as tagging units, while literatures *(Zhang et al., 2006; Zhao and Kit, 2007a; Zhang and Clark, 2007)* focus on employing lexical words or subwords as tagging units."

- "A number of statistical parsing models have recently been developed for CCG and used in parsers applied to newspaper text *(Clark, Hockenmaier, and Steedman 2002; Hockenmaier and Steedman 2002b; Hockenmaier 2003b)*."

3.2 Data Annotation for Citation Classification

This section describes a classifier that is used to distinguish between constituent and parenthetical citations. A classifier is needed because heuristic-based approaches can fall short, as we will see in Section 3.3.2. Citation styling varies throughout different journals and conferences; some styles use citations in brackets (e.g., "Smith [2000]" and "[Smith, 2000]"), numerical citations (e.g., "[1]"). There is no standard set of rules by which an author uses citations, and as a result the way citations are used by authors vary. Some authors use either CCs or PCs exclusively; some may always use CCs with a preposition (e.g., "..., as shown by Smith (2000)."), whereas others may use CCs with a verb (e.g., "We follow Smith (2000), by ..."). A classifier performs better than a heuristics-based approach in applying citation classification to other scientific areas and journals, as well as dealing with the different writing styles of authors.

Mechanical Turk was used to annotate citations from the citation sentences of two data sets of scientific documents. The results from the annotations by Mechanical Turk, the results of which are used as training and evaluation data for the classifier.

The data sets that were used for training and evaluating the classifier were drawn from the ACL Anthology Network (Joseph and Radev, 2007) in the research areas of Question Answering (QA) and Dependency Parsing (DP). The two sets of papers were compiled by selecting papers from the ACL Anthology Network that had the words "Question Answering" and "Dependency Parsing," respectively, in

the title and the content of the paper. There were 10 papers in the QA data set and 16 papers in the DP data set.

The citation sentences from these two data sets are used in the Mechanical Turk tasks described next, and are used as training and evaluation data for the citation classifier. Amazon's Mechanical Turk is a web service where anyone can post a simple human computation task, and pay workers on the system (called *Turkers*) are paid to complete them. I used Mechanical Turk to annotate the citations from the DP and QA datasets as being constituent or parenthetical.[1] The results of these annotations are used to train and evaluate the classifier in Section 3.3.

There were three main Turk tasks: a pilot study, a task to identify vague/unclear sentences, and final task to annotate all citations. Each of these tasks is described, in turn, below.

3.2.1 Pilot Study: Human Agreement on Citation Classification

Before initiating more detailed Mechanical Turk tasks, I conducted a pilot study to determine whether Turkers could agree on the citation classification task. In the citation classification task, Turkers were presented with a citation sentence, with a citation highlighted. They were then asked to classify the citation as "constituent", "parenthetical", or "ambiguous/incorrect citation". The "ambiguous/incorrect" choice was used in case our citation detection was erroneous, or if

[1]Note: The terminology presented to Turkers was slightly different from that used in this thesis. For Turkers, constituent citations were called "syntactic" citations, and parenthetical citations were called "non-syntactic" citations. This terminology was more accessible to a Turker, who may not have experience in linguistics.

the Turker was unable determine the category to which the citation belonged.

Turkers annotated 50 citations in 50 different randomly selected citation sentences from the citation texts from QA and DP. Four Turkers were allowed to annotate each citation. Nine different Turkers participated in the pilot study, annotating an average of 22.2 citations each. The Krippendorff (Passonneau et al., 2006) agreement score was 0.786, which I found to be sufficient to continue with the remaining tasks, and sufficient for the main task of annotating all citations in the QA and DP sets to be used as training data for citation classification.

3.2.2 Identify Vague/Unclear Sentences Task

After the pilot study, Turkers were asked to identify any vague/unclear citation sentences that occurred in the DP and QA data sets. I define a vague/unclear sentence as any sentence that contains special symbols/characters from LaTeX (e.g., $\sum$, $\sqrt{x}$, $\prod$), or any other wording or phrasing that isn't coherent. The main goal of this task was to eliminate sentences where citations were not the only source of noise. By doing so, it is guaranteed that the only source of noise in the remaining citation sentences are the citations themselves. In the task, Turkers were presented with a citation sentence, and asked to label it as "clean" or "vague/unclear". Each sentence was annotated by three different Turkers.

Once this task was completed, the QA and DP data sets were updated by removing sentences that were labeled "vague/unclear" by at least two Turkers. In

total, 29 different Turkers participated in the task, annotating an average of 50.1 sentences each. Out of the 484 total citation sentences in the QA and DP sets, 52 were labeled vague/unclear (10.74%). Turkers found this task hardest to agree upon, with a Krippendorff agreement score of 0.469. I attribute this to the task being more open-ended than the other tasks, and perhaps there were not enough examples in quantity or quality provided to help Turkers with the task. In addition, it could also be due to the confusing content and style of ACL papers for a non-specialist reader. However, this annotation task was used as a filter to ensure I studied sentences in which the interference was caused by citations, and not due to other features of the sentences from the ACL Anthology Network (or sentences taken from LaTeX papers). Despite the low agreement score, it was appropriate since the goal of the task is to ensure that the citations are the only source of noise in the citation sentences.

3.2.3 Annotate Citations Task

The final Turk task I conducted was similar to the pilot study, but using the entire set of citation sentences from DP and QA that were identified as being "clean" sentences from the *Identify Vague/Unclear Sentences Task*. Turkers were presented with a citation sentence, wherein a citation was highlighted. The Turkers were then asked to classify the citation as "constituent" or "parenthetical". Each citation was annotated by three different Turkers.

A citation was classified as "constituent" or "parenthetical" if at least two Turkers agreed on the associated labeling. In the task, 30 different Turkers participated, annotating an average of 69 citations each. Out of the 690 citations from the non-vague/unclear sentences, 370 were labeled as parenthetical (53.62%), and 320 were labeled as constituent (46.38%). Similar to the pilot study, the Krippendorff agreement score was 0.752.

3.3 Training a Citation Classifier

The citations labeled by Turkers in Section 3.2 were used in training and evaluating a maxent classifier (Daumé III, 2008). This section describes the feature set used for the classifier, and an evaluation of the classifier with random, one-label, and heuristic-based classifiers as a baseline comparison.

3.3.1 Feature Selection

The feature set used for the classifier is as follows:

- Words and part-of-speech tags of the words before and after a citation in a ± 2 window. For example, consider the citation sentence, "We used bootstrapping *(Abney, 2002)* which refers to a problem setting in which one is given a small set of labeled data and a large set of unlabeled data, and the task is to induce a classifier." Here the words before the citation are "used boostrapping," and the words after are "which refers." If the citation was located at the beginning or end of a sentence, it was indicated with BOS and EOS tags, respectively.

- The type of parenthesis around the citation. The parentheses either surround the year (Type 0, e.g., "Whidby (2012)"), or the parentheses surround the entire citation (Type 1, e.g., "(Whidby, 2012)").

- Whether any punctuation follows the citation (comma, period, semicolon, etc.)

Part of speech tags were obtained using the "wordsAndTags" output format of the Stanford Parser.

3.3.2 Classification Evaluation

The performance of the maxent classifier was compared with two baselines (a random and one-label classifier), and the heuristics-based approach used by Abu-Jbara and Radev (2011). The one-label classifier labeled each citation as CC.

The classifier was evaluated intrinsically on the classification task in two cases. In the first case, the maxent classifier was trained on the labeled citations from the DP data set, and all classifiers were evaluated on the QA data set (referred to as DP train/QA eval, or DP-QA). In the second case, the maxent classifier was trained on the labeled citations from the QA data set, and all classifiers were evaluated on the DP data set (referred to as QA train/DP eval, or QA-DP). The classifiers were evaluated on accuracy, where the label determined by the Turkers from Section 3.2 was considered the true label. The results are presented in Table 3.1 for the DP train/QA eval and QA train/DP eval splits. The maxent classifier trained on the set of features presented in Section 3.3.1 handily outperforms the two baselines and the heuristics-based approach in both cases. This classifier is used as part of the

Classifier Performance: Classification Task		
Classifier	DP-QA	QA-DP
Random	0.44	0.48
One-label	0.58	0.64
AJR	0.64	0.79
Maxent	**0.91**	**0.87**

Table 3.1: Accuracy of various classifiers on citation classifying task for DP train, QA eval (DP-QA) and QA train, DP eval (QA-DP) splits. AJR refers to the heuristics-based approach used in Abu-Jbara and Radev .

citation handling process, which is presented in the following chapter.

Chapter 4

Citation Handling Process

In this chapter, I present my approach to citation handling, a means for pre-processing citations in scientific documents. We will walk through the five steps of the citation handilng process, illustrating the impact of each step on the *example citation sentence* shown in Figure 4.1.

```
Some previous work (Peng et al., 2004; Tseng et al., 2005;
Low et al., 2005) illustrated the effectiveness of using
characters as tagging units, while literatures (Zhang et al.,
2006; Zhao and Kit, 2007a; Zhang and Clark, 2007) focus on
employing lexical words or subwords as tagging units.
```

Figure 4.1: The example citation sentence that will be traced through the citation handling process.

My approach to citation handling is to pre-process each citation in the citation sentence before it is passed to the parser, and then to post-process it afterwards. In pre-processing, the citation is either replaced or removed from the sentence, based on its type. In post-processing, citations that were pre-processed are re-inserted back in to the citation sentences. A variant of these steps are executed to produce a set of sentence compressions using Trimmer (Zajic et al., 2007); specifically, the citation sentences are post-processed after all sentence compressions have been generated.

For pre-processing constituent citations, the entire citation is replaced with the placeholder text "*CITATIONX*", where X is a unique number assigned to the

citation. With Trimmer, the original citation text is re-inserted back into the sentence using the unique number assigned to it, after all compressions for a sentence have been generated. Examples of pre-processing constituent citations are shown below:

- *Before:* "Moreover, the proof relies on lexico-semantic knowledge available from WordNet as well as rapidly formatted knowledge bases generated by mechanisms described in *(Chaudri et al, 2000)*."

  After: "Moreover, the proof relies on lexico-semantic knowledge available from WordNet as well as rapidly formatted knowledge bases generated by mechanisms described in *CITATION1*."

- *Before:* "Some Q&A systems, like *(Moldovan et al, 2000)* relied both on NE recognizers and some empirical indicators."

  After: "Some Q&A systems, like *CITATION2* relied both on NE recognizers and some empirical indicators."

- *Before:* "More details on the memory-based prediction can be found in *Nivre et al (2004)* and *Nivre and Scholz (2004)*."

  After: "More details on the memory-based prediction can be found in *CITATION3* and *CITATION4*."

For pre-processing parenthetical citations, the citation is removed entirely from the sentence. In the case of citation handling post-processing with Trimmer, the parenthetical citations are currently re-inserted at the end of the sentence, after all sentence compressions have been generated. It is difficult to determine what part

of a sentence's parse tree parenthetical citations are associated with; when it comes to re-inserting parenthetical citations with Trimmer, determining what part of the parse tree a parenthetical citation is associated with is crucial in deciding whether to re-insert the citation (since Trimmer may have removed the part of the parse tree the citation is associated with in creating a sentence compression). Further investigation into determining the association of parenthetical citations with parts of a sentence's parse tree is an area for furture work. Examples of pre-processing parenthetical citations are shown below:

- *Before:* "If the expected answer types are typical named entities, information extraction engines *(Bikel et al 1999, Srihari and Li 2000)* are used to extract candidate answers."

  After: "If the expected answer types are typical named entities, information extraction engines are used to extract candidate answers."

- *Before:* "In English as well as in Japanese, dependency analysis has been studied *(Lafferty et al, 1992; Collins, 1996; Eisner, 1996)*."

  After: "In English as well as in Japanese, dependency analysis has been studied."

- *Before:* "That work extends the maximum spanning tree dependency parsing framework *(McDonald et al, 2005a; McDonald et al, 2005b)* to incorporate features over multiple edges in the dependency graph."

  After: "That work extends the maximum spanning tree dependency parsing framework to incorporate features over multiple edges in the dependency

graph."

The citation handling process consists of five steps:

1. Detect Citations
2. Unify Citations
3. Extract Features
4. Classify Citations
5. Handle Citations

The following sections explain each of the different steps of the citation handling process in detail.

4.1 Detect Citations

The first step of the citation handling process is to find the occurrences of citations within the citation sentence. This is done using RefTagger (Abu-Jbara and Radev, 2011), which identifies individual citations using regular expressions, and surrounds then with "REF" SGML tags. The results of running RefTagger on the example citation sentence are presented in Figure 4.2. While the groups of individual citations (I define a group of individual citations as citations that fall within the same set of parentheses) are correctly identified, we are more interested in the entire citation itself. This is explained further and implemented in the next step of the process, "Unify Citations."

```
Some previous work (<REF>Peng et al., 2004</REF>; <REF>Tseng et al.,
2005</REF>; <REF>Low et al., 2005</REF>) illustrated the effectiveness
of using characters as tagging units, while literatures (<REF>Zhang
et al., 2006</REF>; <REF>Zhao and Kit, 2007a</REF>; <REF>Zhang and
Clark, 2007</REF>) focus on employing lexical words or subwords as
tagging units.
```

Figure 4.2: The example citation sentence after being passed through RefTagger. RefTagger finds and tags individual citations in a citation sentence.

4.2 Unify Citations

Dealing with the entire citation rather than the group of individual citations identified by RefTagger is more useful for citation handling. Consider if the group of individual citations in the example citation sentence, as presented in Figure 4.2, were classified as parenthetical citations (and as such were removed from the sentence before parsing). Since the REF tags only cover the names of the author(s) and the year of publication, the parentheses and semicolons would be left in the original sentence. Figure 4.3 shows how the sentence would look if this approach were taken. Clearly, having the leftover parentheses and semicolons in the sentence would not help with parsing. If a group of individual citations were instead unified into a single citation, this problem could be avoided. In the case of our example citation sentence, three individual citations "Peng et al., 2004," "Tseng et al., 2005," and "Low et al., 2005" can be unified into the single citation "(Peng et al., 2004; Tseng et al., 2005; Low et al., 2005)".

In the implementation for unifying citations, the code looks for REF tags that occur together within parentheses. It then surrounds the entire citation (including the parentheses) with a REF tag, and removes all REF tags within the parentheses

```
Some previous work ( ; ; ) illustrated the effectiveness
of using characters as tagging units, while literatures ( ;
; ) focus on employing lexical words or subwords as
tagging units.
```

Figure 4.3: How the sentence would look if only the individual citations were removed. It is better to unify the citations into a single group such that the parenthesis and semicolons can also be removed.

```
Some previous work <REF>(Peng et al., 2004; Tseng et al.,
2005; Low et al., 2005)</REF> illustrated the effectiveness
of using characters as tagging units, while literatures
<REF>(Zhang et al., 2006; Zhao and Kit, 2007a; Zhang and
Clark, 2007)</REF> focus on employing lexical words or
subwords as tagging units.
```

Figure 4.4: The example citation sentence after having groups of individual citations unified into a single citation.

(i.e., the original REF tags from the individual citations). Figure 4.4 shows the example citation sentence after the citations have been unified. All the groups of individual citations have now been unified into single citations.

4.3 Extract Features

After the citations have been unified, the next step in the process is to extract features from the sentence to pass into the citation classifier. The features used for the classifer were presented earlier in Section 3.3, but the extraction of these features from a citation sentence is covered in depth here.

4.3.1 Parentheses Type

The first feature that is determined is the type of parentheses surrounding the citation. A "Type 0" parentheses is where the parentheses surround the entire citation (e.g., "(Whidby, 2012)"), and a "Type 1" parentheses is where the parentheses surround the year in the citation (e.g., "Whidby (2012)"). Figure 4.4 shows the example citation sentence with unified REF tags. In both cases, the citations in the REF tags have "Type 0" parenthesis.

4.3.2 Words and Tags

The next step is to determine the tags of the words before and after the citations, in a ± 2 window. In the case of the example citation sentence, this would be the words "previous," "work," "illustrated," and "the" for the first citation, and the words "while," "literatures," "focus," and "on" for the second citation. To determine the tags of the words, the citation sentence is fed into the Stanford Parser using the "wordsAndTags" output option, with the citations temporarily replaced with the filler text "CITATION-X-Y," where X is a unique identifier for the citation and Y is the type of parenthesis determined from Section 4.3.1 (0 or 1). Figure 4.5 shows the example citation sentence formatted for input into the Stanford Parser, and Figure 4.6 shows the output from the Stanford Parser using the "wordsAndTags" option. In the case of the first citation, the tags for "previous," "work," "illustrated," and "the" are "JJ," "NN," "VBD," and "DT," respectively.

```
Some previous work CITATION-1-0 illustrated the effectiveness
of using characters as tagging units, while literatures
CITATION-2-0 focus on employing lexical words or
subwords as tagging units.
```

Figure 4.5: The example citation sentence as it is fed into the Stanford Parser to determine the tags of the words before and after the citations.

```
Some/DT previous/JJ work/NN CITATION-1-0/NN illustrated/VBD
the/DT effectiveness/NN of/IN using/VBG characters/NNS as/IN
tagging/VBG units/NNS ,/, while/IN literatures/NNP
CITATION-2-0/NNP focus/VBP on/IN employing/VBG lexical/JJ
words/NNS or/CC subwords/NNS as/IN tagging/JJ units/NNS ./.
```

Figure 4.6: The output from the Stanford Parser using the "wordsAndTags" option with the example citation sentence.

4.3.3 Punctuation

The final feature for the classifier that is extracted from the sentence is whether or not punctuation follows the citation. This punctuation could be a comma or semicolon following the citation, or a period denoting the end of the sentence. If there is punctuation, then the value of this feature is 1, otherwise it is 0. In the case of the example citation sentence, both citations do not have punctuation, and thus labeled as 0.

4.4 Classify Citations

After the features have been extracted from the citation sentence, it is classified as being a constituent or parenthetical citation by the maxent classifier described previously in Chapter 3. A classification of "1" declares a citation to be constituent, while "0" declares the citation to be parenthetical. The REF tag of the citation

```
Some previous work <REF type="PC">(Peng et al., 2004;
Tseng et al., 2005; Low et al., 2005)</REF> illustrated
the effectiveness of using characters as tagging units,
while literatures <REF type="PC">(Zhang et al., 2006;
Zhao and Kit, 2007a; Zhang and Clark, 2007)</REF> focus
on employing lexical words or subwords as tagging units.
```

Figure 4.7: The example citation sentence after the citations have been classified. Both citations have been classified as parenthetical citations, and as such are labeled with type "PC."

```
Some previous work illustrated the effectiveness of
using characters as tagging units, while literatures
focus on employing lexical words or subwords as
tagging units.
```

Figure 4.8: The example citation sentence after the classified citations have been handled. Since both citations were classified as type "PC," they are removed from the sentence before parsing.

is then updated with a "type" attribute to reflect the citation's type, with "CC" and "PC" used as attribute values to denote constituent citations and parenthetical citations, respectively. Figure 4.7 shows the example citation sentence after classification. Both citations were classified as being parenthetical citations.

4.5 Handle Citations

The final step in the process is to handle the citations. Recall that constituent citations are replaced with a filler text, and parenthetical citations are removed from the sentence before being passed on to a parser. Figure 4.8 shows the example citation sentence in its final stage after citation handling, and is the sentence that will be used for parsing. Since both citations were labeled as "PC," they are both removed.

This chapter presented the data and Mechanical Turk tasks that were used to train and evaluate a citation classifier. The citation classifier is used as part of the citation handling process, which pre-processes citations in five steps: Detect Citations, Unify Citations, Extract Features, Classify Citations, and Handle Citations. The next chapter examines the effects citation handling has on the behavior of the Stanford Parser, a sentence compression system (Trimmer), and a multi-document summarization system (MASCS).

Chapter 5

Application of Citation Handling to Adapt Generic NLP Tools to Scientific Literature

This chapter examines the application of citation handling to three generic NLP tools: the Stanford Parser, Trimmer (a sentence compressor), and MASCS (a multidocument summarization system). For each NLP tool, specific examples will be presented in which citation handling improves the output of the tool. In examining citation handling's effect on Trimmer, we will revisit examples from previous chapters for the purpose of illustrating the effect of citation handling on all three NLP tools.

5.1 Stanford Parser

In this section, the erroneous parse trees created by the Stanford Parser (Klein and Manning, 2003) discussed in Section 1.1.1 are presented again for convenience in Figures 5.1 and 5.3. We will demonstrate the application of citation handling for improving the quality of the parse trees. The parse trees of citation sentences that have been pre-processed using citation handling are compared to those that have not been pre-processed. Citation handling is shown to improve the quality of the parse trees generated by the Stanford Parser.

Consider the citation sentence and its corresponding parse tree in Figure 5.1,

which was created by the Stanford Parser without citation handling. This parse tree has several issues: both the first citation, the "(CC and)" conjunction, and the second citation should be attached under the PP in "(VP (VBN discussed) (PP (IN in)))." In addition, the PP has been closed off too early. Figure 5.2 shows the parse tree of the same sentence, except in this case the citations have been preprocessed with citation handling. With citation handling, all the issues with the bad parse tree have been fixed - the two citations and the conjunction joining them are now attached under PP, and the PP has been closed off appropriately.

Also consider the parse tree of the citation sentence parsed without citation handling presented in Figure 5.3, which also contains numerous errors. The conjunction "and/or" and the adjective "linear" should be attached to the ADJP to which the other adjective "deterministic" is attached. In addition, the adjective "linear" has been tagged as a verb in a verb phrase with the citation. Figure 5.4 presents the parse tree of the same sentence, except the citations have been pre-processed with citation handling. Again, all the errors have been fixed as a result of citation handling. "Linear" has been correctly tagged as an adjective, and both it and the conjunction "and/or" have been correctly placed in the ADJP.

This section has examined specific examples where the parse trees produced by the Stanford Parser are improved as a result of citation handling. These parse trees are used in Trimmer, a sentence compressor, to apply rules to the parse tree to generate sentence compressions. The next section introduces Trimmer, and examines what effect these improved parse trees, a result of citation handling, has on Trimmer's sentence compressions.

```
...
(, ,)
(NP (PRP we))
(VP (VBD revisited)
  (NP
    (NP (NP
        (NP (DT the) (NN test) (NNS cases))
        (VP (VBN discussed)
          (PP (IN in))))
      (PRN (-LRB- -LRB-)
        (NP (NNP Carroll)
          (CC et)
          (NNP al))
        (, ,)
        (NP (CD 1999))
        (-RRB- -RRB-)))
    (CC and)
    (SBAR
      (S
        (VP
          (PRN (-LRB- -LRB-)
            (NP (NNP Koller)
              (CC and)
              (NNP Striegnitz))
            (, ,)
            (NP (CD 2002))
            (-RRB- -RRB-))
          (PP (IN by)
            (NP
              (NP (JJ producing) (JJ similar) (NNS sentences))
              (PP (IN in)
                (NP (NNP French)))))))))
(. .)))
```

Figure 5.1: The parse tree for the citation sentence *"To get an estimate of how our realiser compares with existing published results, we revisited the test cases discussed in [Carroll et al, 1999] and [Koller and Striegnitz, 2002] by producing similar sentences in French."* without using citation handling.

5.2 Trimmer

Trimmer (Zajic et al., 2007) is a linguistically-motivated, heuristics-based approach to sentence compression. It applies syntactic compression rules (also called

```
...
(, ,)
  (NP (PRP we))
    (VP (VBD revisited)
      (SBAR
        (S
          (NP (DT the) (NN test) (NNS cases))
            (VP (VBD discussed)
              (PP (IN in)
                (NP (NNP CITATION1)
                  (CC and)
                (NNP CITATION2)))
              (PP (IN by)
                (S
                  (VP (VBG producing)
                    (NP (JJ similar) (NNS sentences))
                    (PP (IN in)
                      (NP (NNP French)))))))))
(. .)))
```

Figure 5.2: The parse tree tree for the citation sentence *"To get an estimate of how our realiser compares with existing published results, we revisited the test cases discussed in [Carroll et al, 1999] and [Koller and Striegnitz, 2002] by producing similar sentences in French."* when using citation handling.

Trimmer rules) to a parse tree generated by the Stanford Parser. These Trimmer rules mask nodes in the tree - if a node in the parse tree is marked as being masked, then its leaf node descendes do not appear in the string representation of that sentence compression candidate. For example, one Trimmer rule is the conjunction rule, where a conjunction containing two children will be split into three compressions: one containing the original text, one containing the first child only, and one containing the second child only.

Post-processing of citations is done after all sentence compression candidates have been generated. As a reminder, during pre-processing, constituent citations

```
(ROOT
  (S
    (ADVP (RB Recently))
    (NP (JJ statistical) (JJ dependency) (NN parsing) (NNS techniques))
    (VP (VBP have)
      (VP (VBN been)
        (VP
          (VP (VBN proposed)
            (SBAR
              (WHNP (WDT which))
              (S
                (VP (VBP are)
                  (ADJP (JJ deterministic))))))
          (CC and/or)
          (VP (VBN linear)
            (PRN (-LRB- -LRB-)
              (NP
                (NP (NNP Yamada)
                  (CC and)
                  (NNP Matsumoto))
                (, ,)
                (NP (CD 2003))
                (, ;)
                (NP (NNP Nivre)
                  (CC and)
                  (NNP Scholz))
                (, ,)
                (NP (CD 2004)))
              (-RRB- -RRB-))))))
    (. .)))
```

Figure 5.3: The parse tree for the citation sentence *"Recently statistical dependency parsing techniques have been proposed which are deterministic and/or linear (Yamada and Matsumoto, 2003; Nivre and Scholz, 2004)."* created without using citation handling.

are replaced with a filler text containing a unique identifier (e.g., "CITATION-24," where 24 is a unique ID number). Information on the constituent citations is stored in a hash table, where the unique identifier is the key and the original citation is the value. During post-processing, any unique identifiers in the candidate sentences

```
(ROOT
  (S
    (ADVP (RB Recently))
    (NP (JJ statistical) (JJ dependency) (NN parsing) (NNS techniques))
    (VP (VBP have)
      (VP (VBN been)
        (VP (VBN proposed)
          (SBAR
            (WHNP (WDT which))
            (S
              (VP (VBP are)
                (ADJP (JJ deterministic)
                  (CC and/or)
                  (JJ linear))))))))
    (. .)))
```

Figure 5.4: The parse tree for the citation sentence *"Recently statistical dependency parsing techniques have been proposed which are deterministic and/or linear (Yamada and Matsumoto, 2003; Nivre and Scholz, 2004)."* created using citation handling.

are replaced with their associated original citation.

When pre-processing parenthetical citations, the citation is removed from the sentence entirely. Each citation that is removed is added to a list associated with that sentence. During post-processing, the list of removed citations for that sentence is combined into a single citation. For example, the citations "(Smith, 2010)" and "(Williams, 2011)" are combined into a single citation, "(Smith, 2010; Williams, 2011)." This is the current approach to re-inserting parenthetical citations since the location of these citations in the original citation sentence are not stored. A better means of re-inserting the parenthetical citations back into the sentence is left as future work.

The sentence compression candidates created from Trimmer are used as part of a summarization system, MASCS. The summaries generated from MASCS are

```
Some previous work (Peng et al., 2004; Tseng et al., 2005;
Low et al., 2005) illustrated the effectiveness of using
characters as tagging units, while literatures (Zhang et al.,
2006; Zhao and Kit, 2007a; Zhang and Clark, 2007) focus on
employing lexical words or subwords as tagging units.
```

Figure 5.5: The example citation sentence that will be traced through this chapter.

used for an extrinsic evaluation of citation handling in Sections 6.4.2 and 6.4.3.

5.2.1 Effect of Citation Handling on Trimmer

As a result of citation handling causing the Stanford parser to generate better parse trees, Trimmer should be able to create better sentence compression candidates. In this section, two examples of the effect of citation handling on Trimmer are presented.

In the first example, Trimmer is run on the example citation sentence used throughout Chapter 4, and presented again for convenience in Figure 5.5. Without citation handling, Trimmer creates 96 sentence compression candidates from the example citation sentence, many of which are exactly the same except for differences in the citations. Since the example citation sentence has two "and" citations, "Zhao and Kit, 2007a" and "Zhang and Clark, 2007," Trimmer will apply the conjunction rule to both. Figure 5.6 shows eight sentence compressions that are exactly the same, except for differences in the text of the citations. Specifically, the sentence compressions vary in the different combinations of the citations "Zhao and Kit, 2007a" and "Zhang and Clark, 2007."

On the other hand, as a result of having the "and" citations removed when

using citation handling, Trimmer creates 12 sentence compression candidates. This shows that without citation handling, Trimmer can have an exponential growth in the number of sentence compression candidates just because of "and" citations. Since the extra compression candidates generated without citation handling are essentially the same, this means wasted computation time for Trimmer, as well as wasted computation time for any system that uses the sentence compressions from Trimmer, such as MASCS, a summarization system.

For the second example, Trimmer is run on the example citation sentence that was shown to achieve a better parse tree with citation handling, "To get an estimate of how our realiser compares with existing published results, we revisited the test cases discussed in [Carroll et al, 1999] and [Koller and Striegnitz, 2002] by producing similar sentences in French." Recall that in the analysis from Section 5.1, the parse tree misplaced the "(CC and)" separating the two citations when no citation handling was used, and was placed correctly with citation handling. Figure 5.7 presents some sentence compressions that were generated with and without citation handling. The first three candidates were generated with the bad parse tree that resulted from not handling citations. Any compression candidate that had Trimmer's conjunction rule applied to the conjunction separating the citations now removed the entire phrase "[Koller and Striegnitz, 2002] by producing similar sentences in French." The last three candidates in Figure 5.7 were generated with the better parse tree as a result of citation handling (the better parse tree was presented in Figure 5.2 in Section 5.1). Here, only the citation "[Koller and Striegnitz, 2002]" is removed, and the phrase "by producing similar sentences in French" remains in

- Some previous work (Peng et al., 2004; Tseng et al., 2005; Low et al., 2005) illustrated the effectiveness of using characters as tagging units, while literatures (Zhang et al., 2006; Zhao, 2007a; Zhang and Clark, 2007) focus on employing lexical words or subwords as tagging units.

- Some previous work (Peng et al., 2004; Tseng et al., 2005; Low et al., 2005) illustrated the effectiveness of using characters as tagging units, while literatures (Zhang et al., 2006; Kit, 2007a; Zhang and Clark, 2007) focus on employing lexical words or subwords as tagging units.

- Some previous work (Peng et al., 2004; Tseng et al., 2005; Low et al., 2005) illustrated the effectiveness of using characters as tagging units, while literatures (Zhang et al., 2006; Zhao and Kit, 2007a; Zhang, 2007) focus on employing lexical words or subwords as tagging units.

- Some previous work (Peng et al., 2004; Tseng et al., 2005; Low et al., 2005) illustrated the effectiveness of using characters as tagging units, while literatures (Zhang et al., 2006; Zhao and Kit, 2007a; Clark, 2007) focus on employing lexical words or subwords as tagging units.

- Some previous work (Peng et al., 2004; Tseng et al., 2005; Low et al., 2005) illustrated the effectiveness of using characters as tagging units, while literatures (Zhang et al., 2006; Zhao, 2007a; Zhang, 2007) focus on employing lexical words or subwords as tagging units.

- Some previous work (Peng et al., 2004; Tseng et al., 2005; Low et al., 2005) illustrated the effectiveness of using characters as tagging units, while literatures (Zhang et al., 2006; Zhao, 2007a; Clark, 2007) focus on employing lexical words or subwords as tagging units.

- Some previous work (Peng et al., 2004; Tseng et al., 2005; Low et al., 2005) illustrated the effectiveness of using characters as tagging units, while literatures (Zhang et al., 2006; Kit, 2007a; Zhang, 2007) focus on employing lexical words or subwords as tagging units.

- Some previous work (Peng et al., 2004; Tseng et al., 2005; Low et al., 2005) illustrated the effectiveness of using characters as tagging units, while literatures (Zhang et al., 2006; Kit, 2007a; Clark, 2007) focus on employing lexical words or subwords as tagging units.

Figure 5.6: Eight citation sentence compressions from Trimmer that were created without the use of citation handling. Each sentence is exactly the same except for minor differences in the citations as a result of applying the conjunction Trimmer rule.

Original Sentence: "To get an estimate of how our realiser compares with existing published results, we revisited the test cases discussed in [Carroll et al, 1999] and [Koller and Striegnitz, 2002] by producing similar sentences in French." Without Citation Handling

1. To get an estimate of how our realiser compares with existing published results, we revisited the test cases discussed in [Carroll et al, 1999].
2. To get an estimate, we revisited the test cases discussed in [Carroll et al, 1999].
3. To get an estimate of how our realiser compares, we revisited the test cases discussed in [Carroll et al, 1999].

With Citation Handling

1. To get an estimate of how our realiser compares with existing published results, we revisited the test cases discussed in [Carroll et al, 1999] by producing similar sentences in French.
2. To get an estimate, we revisited the test cases discussed in [Carroll et al, 1999] by producing similar sentences in French.
3. To get an estimate of how our realiser compares, we revisited the test cases discussed in [Carroll et al, 1999] by producing similar sentences in French.

Figure 5.7: Examples of sentences generated with and without citation handling for the citation sentence "To get an estimate of how our realiser compares with existing published results, we revisited the test cases discussed in [Carroll et al, 1999] and [Koller and Striegnitz, 2002] by producing similar sentences in French." Without citation handling, the conjunction rule removes the whole phrase "[Koller and Striegnitz, 2002] by producing similar sentences in French."

the compressions. Without citation handling, Trimmer can unintentionally remove entire phrases from sentence compressions as a result of bad parse trees.

Since Trimmer is able to generate higher quality (and less redundant) sentence compressions with citation handling, MASCS should also be able to generate higher quality summaries. In the next section, MASCS is introduced in detail, followed by an examination of the effects of citation handling on the quality of MASCS summaries.

5.3 MASCS - Multiple Alternate Sentence Compression Summarizer

MASCS (Zajic et al., 2007) is a summarization system that utilizes Trimmer's sentence compression candidates to create summaries for a single or set of documents (referred to as a *cluster*). These documents could be news articles, scientific documents, etc. Summarization with MASCS is performed in three stages. In the first stage, Trimmer generates several compressed sentence candidates for every sentence in a document from the cluster. The second stage involves calculating various ranking features for each of the compressed sentence candidates. In the final stage, sentence candidates are chosen for inclusion in the summary, and are chosen based on a linear combination of features.

There are eight different features used for ranking candidate sentences for summarization in MASCS, broken into two categories: fixed features and dynamic features. The fixed features are computed once for each candidate sentence, and the dynamic features are computed every time a sentence is added to the summary.

The fixed features are:

1. *Position* - The zero-based position of the sentence in the document.

2. *Sentence Relevance* - The relevance of the sentence to the query (if a query is provided).

3. *Document Relevance* - The relevance of the sentence's document to the query (if a query is provided).

4. *Sentence Centrality* - The centrality score of the sentence to the sentence's

document.

5. *Document Centrality* - The centrality score of the sentence's document to the cluster.

6. *Trims* - The number of Trimmer rules applied to the sentence (can be weighted based on type of Trimmer rule applied).

The dynamic features are:

1. *Redundancy* - The measure of how similar the sentence is to the current sentences in the summary.

2. *Sent-from-doc* - The number of sentences already selected for the summary from the sentence's document.

The final score assigned to a candidate sentence is a linear combination of these features. The final score for the candidate sentence is then used in the Sentence Selection stage to choose sentences for the summary.

Sentences are selected to be used in the summary based on their final score from the Ranking Features, and Maximal Marginal Relevance (Carbonell and Goldstein, 1998). The summaries generated by MASCS is used for an extrinsic evaluation of citation handling in Sections 6.4.2 and 6.4.3.

5.4 Effect of Citation Handling on MASCS

With better quality Trimmer sentence compression candidates, MASCS is able to produce better summaries. Figure 5.8 presents a summary created without cita-

tion handling, while Figure 5.9 presents a summary created with citation handling. In the summary in Figure 5.8 that was created without citation handling, a sentence compression from the citation sentence examined in Sections 5.1 and 5.2.1 with the misplaced conjunction has made it into the final summary ("with existing published results, we revisited the test cases discussed in (Carroll et al, 1999)."). On the other hand, the summary created with citation handling presented in Figure 5.9 contains a sentence compression that results from the better quality parse tree and set of Trimmer compressions provided by citation handling, "with existing published results, we revisited test cases discussed in (Carroll et al, 1999) and (Koller and Striegnitz, 2002) by producing similar sentences in French.".

This chapter has presented specific examples of how citation handling improves three NLP components: the Stanford Parser, Trimmer, and MASCS. With the Stanford Parser, better quality parse trees were generated with citation handling. Trimmer was able to avoid creating redundant sentence compressions caused by "and" citations, and the better parse trees resulted in better sentence compressions. Finally, with the better sentence compressions, MASCS was able to generate improved summaries. In the next chapter, evaluations of citation handling are performed on these same three components.

Hahn & Adriaens (1994) ubiquitous requirement of enhanced efficiency of implementations, its inherent
potential for fault tolerance and robustness, and flavor of cognitive plausibility based on psycholinguistic evidences from architecture of human language processor.
Dependency-based statistical language modeling and analysis have also become quite popular.
Nivre (2004) developed history-based learning model.
Y&M 2003 is SVM-shift - reduce parsing model of Yamada and Matsumoto (2003) with existing published results, we revisited the test cases discussed in (Carroll et al, 1999).
In English as well as in Japanese, dependency analysis has been studied e.g., Lafferty et al, 1992; Collins, 1996; Eisner, 1996.
is true of widely used link grammar parser for English (Sleator and Temperley, 1993), which uses dependency grammar of sorts, probabilistic dependency parser of Eisner (1996), and more recently proposed deterministic dependency parsers (Yamada and Matsumoto, 2003; Nivre et al, 2004).
Dependency-based statistical language modeling and parsing have also become quite popular.
Br6kcr, Hahn & Schacht (1994) for more comprehensive treatment considers dependency relations between words as fundamental notion of lingnistic analysis.
Eisner 1996b originally used POS tags to smooth generative model in way.
More details on memory-based prediction can be found in Nivre et al (2004) and Nivre and Scholz (2004).
Schacht et al 1994; Hahn et al 1994.
paper treats resolution of anaphora within framework

Figure 5.8: MASCS summary generated without citation handling.

inverse transformation can also be carried out on test tree (Nivre and Nilsson, 2005; Nivre et al, 2006).
Nivre and Nilsson (2005) improve parsing accuracy for MaltParser by projectivizing training data and applying inverse transformation to output of parser, while Hall and Novak (2005) apply post-processing to output of Charniaks parser (Charniak, 2000).
search for best parse can then be formalized as search for maximum spanning tree (MST) (McDonald et al, 2005b).
For handling nonprojective relations, Nivre and Nilsson (2005) suggested applying pre-processing step to dependency parser, which consists in lifting nonprojective arcs to their head repeatedly, until tree becomes pseudo-projective.
We also intend to use Turkish Treebank, as resource to extract statistical information along lines of Frank et al (2003) and ODonovan et al (2005).
Recently statistical dependency parsing techniques have been proposed which are deterministic and or linear (Yamada and Matsumoto, 2003; Nivre and Scholz, 2004).
Nivre and Scholz (2004) developed history-based learning model.
For details on CoNLL-X shared task and measurements see (Buchholz, et al 2006).
graph shows average 4 report numbers for undirected dependencies on Chinese Treebank 3.0 (Wang et al, 2005).
ubiquitous requirement of enhanced efficiency of implementations, its inherent potential for fault tolerance and robustness, and flavor of cognitive plausibility based on psycholinguistic evidences from architecture of human language processor (Hahn and Adriaens (1994)).
with existing published results, we revisited test cases discussed in (Carroll et al, 1999) and (Koller and Striegnitz, 2002) by producing similar sentences in French.
Dependency-based statistical language modeling and analysis have also become quite popular in statistical natural language processing (Lafferty et al, 1992; Eisner, 1996; Chelba and et al, 1997).

Figure 5.9: MASCS summary generated with citation handling.

Chapter 6

Evaluation

The effect of citation handling is evaluated extrinsically on three NLP systems: the Stanford Parser, Trimmer, and MASCS. For the Stanford Parser, the parser confidence scores are evaluated, in addition to the amount of time it takes the parser to produce parse trees. For Trimmer, the sentence compression candidates produced with and without citation handling are evaluated with a language model. Finally, the summaries produced by MASCS with and without citation handling are evaluated using two standard summarization measures.

6.1 Data

Throughout this chapter, evaluations are performed on six different data sets taken from the ACL Anthology Network (Joseph and Radev, 2007). These data sets were on the topics of Dependency Parsing (DP), Question Answering (QA), Multi-document Summarization (MDS), Semi-supervised Learning (SSL), Conditional Random Fields (CRF), and Wikipedia (Wiki). The data sets were generated by searching for documents

6.2 Effect of Citation Handling on Parsing

We will first evaluate the effect of citation handling on the Stanford Parser. Two evaluations are performed: one on parser confidence scores, and the other on the amount of time taken to produce a parse tree.

6.2.1 Confidence Scores

The first evaluation of citation handling was on the confidence scores of the Stanford Parser.[1] The intuition is that the parser gives higher confidence scores to better quality parses, so if the parser is generally giving higher confidence scores it is generally producing better parses. Figure 6.1 shows the distribution of the confidence scores from the Stanford Parser with and without citation handling. The data appears to be normal and bimodal, with a set of outliers that were much lower in scores. I excluded scores below the threshold of -750, which were considered outliers (indicated by the vertical dark grey line in Figure 6.1). In the no citation handling case 1.17% of the scores were outliers and 2.8% of the scores were outliers in the citation handling case. I ran a Chi-squared test with Yates' continuity correction and found that there was no significant difference in the number of outliers between the conditions.

I conducted a T-test on the scores, and only included sentences whose scores were above the threshold of -750 in both the citation handling and no citation handling cases. The number of sentences where neither condition produced an

[1]The meaning and derivation of these confidence scores from the Stanford Paresr are not explicitly known, and a further investigation is left for future work.

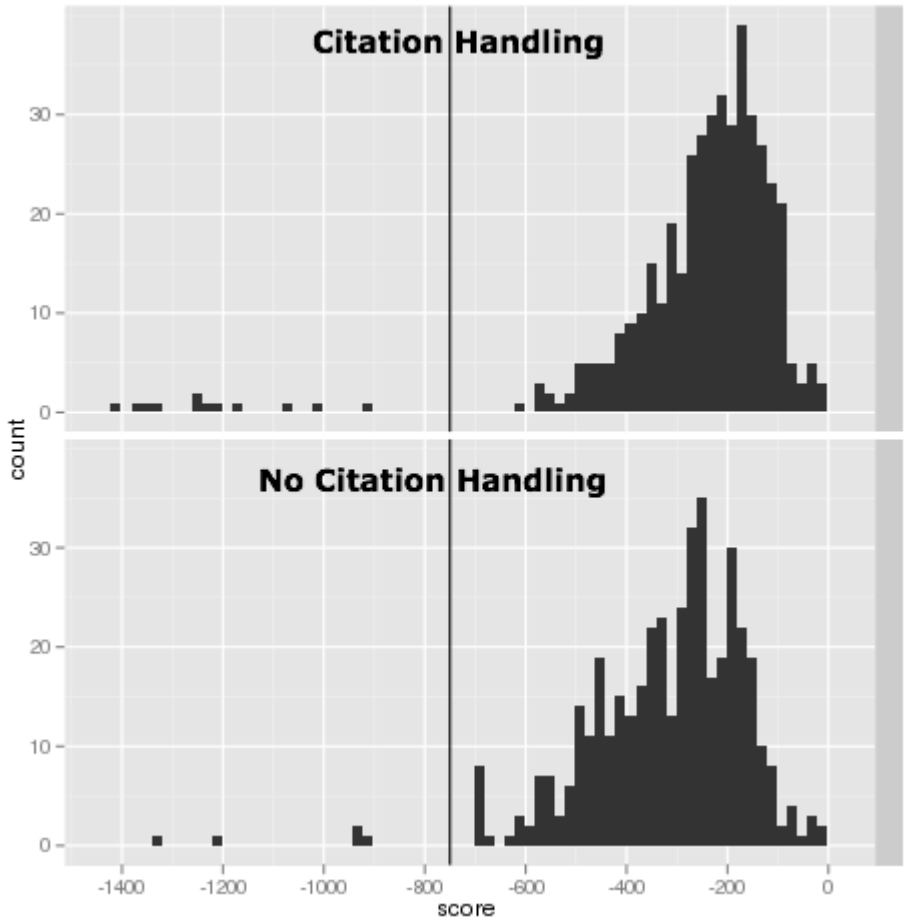

Figure 6.1: Distribution of Stanford Parser confidence scores for citation sentences with and without citation handling. The top half shows scores on sentences with citation handling, and the bottom half shows scores on sentences without citation handling. The dark grey vertical line indicates the threshold for outliers.

outlier was 412 (96.26%). The results of a paired T-test on the confidence scores of the citation sentences found citation handling to have a significant effect, with $p < 0.01$.

6.2.2 Parser Performance

In addition to the confidence scores, I evaluated the time it takes the Stanford Parser to produce parse trees for 100 citation sentences, both with and without

Stanford Parser Performance (seconds)		
	No-CH	**CH**
Run1	265.561s	**71.37s**
Run2	265.281s	**71.003s**
Run3	263.319s	**70.727s**
Run4	265.933s	**70.952s**
Run5	265.902s	**71.081s**
Avg	265.199s	**71.027s**

Table 6.1: Time in seconds for the Stanford Parser to produce parse trees for 100 citation sentences randomly selected from the DP and QA datasets. No-CH indidates that no citation handling was used on the citation sentences, and CH indicates that citaiton was used on the citation sentences.

citation handling. The citation sentences were randomly selected from the DP and QA datasets. Time was measured using the Unix "time" command; specifically, the "real" time output from the "time" command. The tests were run on a MacBook Pro with a 2.53 GHz Intel Core i5 processor and 4 GB of RAM. Table 6.1 presents the results of five runs of the Stanford Parser on the citation sentences. Using citation handling greatly improves the performance of the Stanford Parser, generating parse trees 3.73 times faster than the no citation handling case. Having an almost four times improvement in the time for the parser to produce parse trees is drastic: it would suggest that parenthetical phrases trip up the Stanford Parser. In addition, the syntax for citations is different than "normal" language since it involves the listing of authors' names and a date of publication. A further investigation into all parenthetical phrases (not just citations) would be an interesting avenue for future work.

6.3 Effect of Citation Handling on Sentence Compression

To evaluate the impact of citation handling on Trimmer, the quality of sentence compression candidates generated by Trimmer was evaluated using a language model, because this supports the provision of sentence-trimmed candidates for summarization. The evaluation was done on the Dependency Parsing (DP), Question Answering (QA), Conditional Random Fields (CRF), Semi-supervised Learning (SSL), Multi-document Summarization (MDS), and Wikipedia data sets.

SRILM (Stolcke, 2002) was used to create the language model. The language model was trained on all citation sentences in the ACL Anthology, excluding the sentences that are contained in the evaluation data sets themselves. The sentence candidates were evaluated on trigrams. The score that is reported is an average perplexity-per-word score, which is defined as (6.1):

$$s = 2^{\text{calc}} \tag{6.1}$$

Where "calc" is the log probability divided by the number of words in the sentence (6.2):

$$calc = \frac{\text{logprob}}{\text{num words}} \tag{6.2}$$

The results for DP, QA, CRF, SSL, MDS, and Wikipedia are presented in Figure 6.2. Lower scores indicate that the sentence candidates are of higher quality. In all cases, sentence compressions with citation handling score better than those

without citation handling. The scores for sentence compressions without citation handling increase as the length of the sentence increases. This occurs because as sentence length increases, the number of citations in the sentence also increase, which (without citation handling) results in more opportunities for the noise caused by citations to affect the parse trees and sentence compressions.

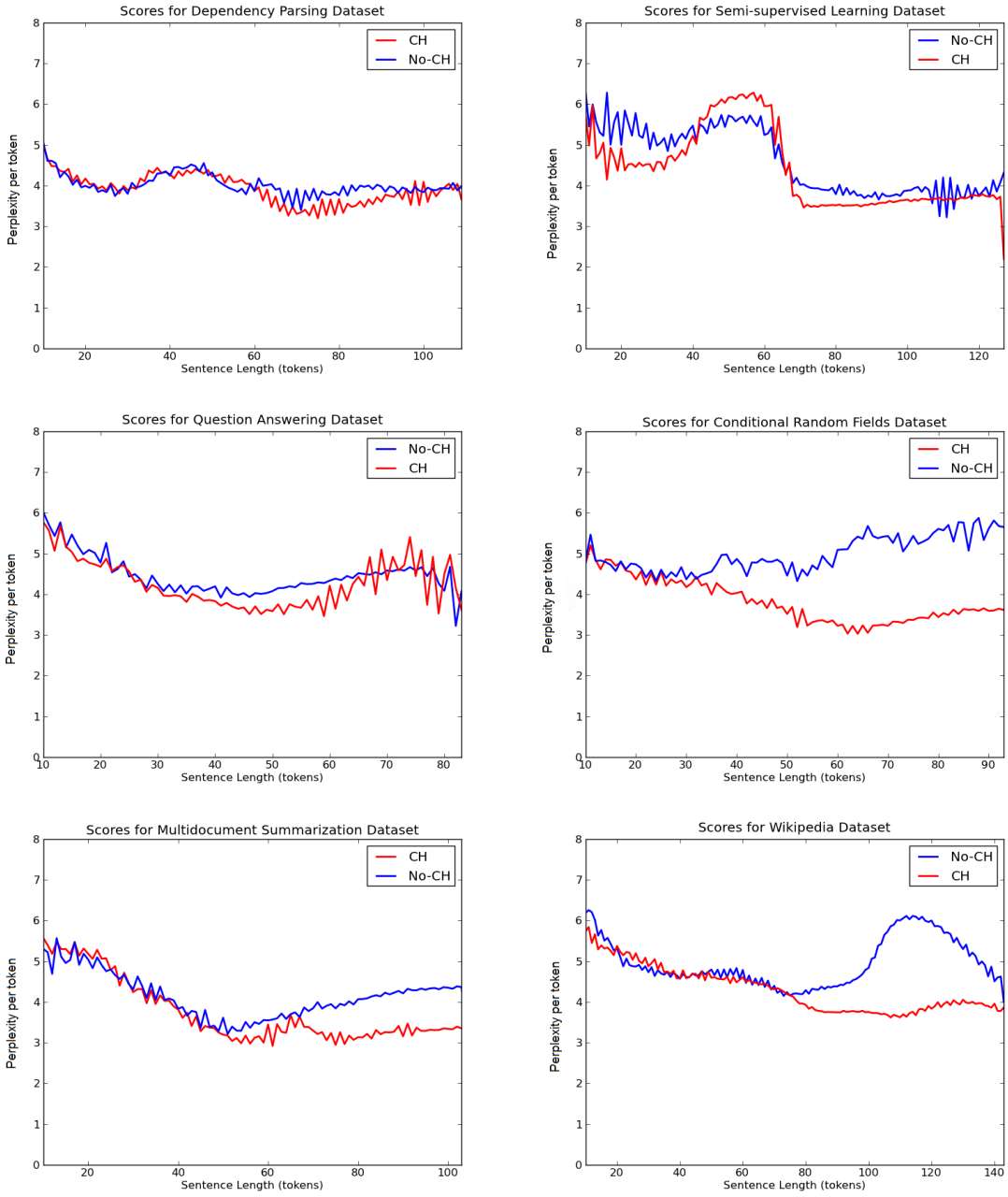

Figure 6.2: Perplexity per token scores for Trimmer sentence compressions for Dependency Parsing (DP), Question Answering (QA), Multi-document Summarization (MDS), Semi-supervised Learning (SSL), Conditional Random Fields (CRF), and Wikipedia (Wiki).

6.4 Effect of Citation Handling on Summarization

To evaluate the impact of citation handling on MASCS, two standard summarization meaures, ROUGE (Lin, 2004) and Pyramid (Lin and Demner-Fushman,

2006; Nenkova and Passonneau, 2004; Hildebrandt et al., 2004; Voorhees, 2003), were used. In both cases, summaries were generated with and without citation handling, and are compared to a baseline random summary.

6.4.1 Gold Standard Summaries

In addition to the gold standard summaries that were genereated for the DP and QA datasets in Mohammad et al. (2009), eleven fluent English speakers were tasked with creating 250-word summaries for Conditional Random Fields, Multi-document Summarization, Semi-supervised Learning, and Wikipedia data sets. At least four human summaries were generated for each data set.

These human summaries are used as gold standard summaries in the ROUGE evaluations to determine how well MASCS performed with and without citation handling.

6.4.2 ROUGE

Table 6.2 presents ROUGE scores of each of the human-generated 250-word surveys against each other for DP and QA using jackknifing. Table 6.3 shows the ROUGE scores of each of the human-generated summaries for the other datasets. The average (last column) is what the automatic surveys can aim for. Each of the the surveys generated by two variants of MASCS (one with citation handling, one without) were evaluated against the references. Table 6.4 lists ROUGE scores of surveys when the manually created 250-word survey of the various citation texts

were used as reference summaries. Among the automatic summarizers, MASCS-CH, the version of MASCS with citation handling, performs best for every data set. Figure 6.3 presents the ROUGE scores with 95% confidence intervals for the six data sets. For DP, MDS, and CRF, the lower bound of the 95% confidence interval for summaries created with citation handling lies above the ROUGE-2 scoers of summaries created without citation handling.

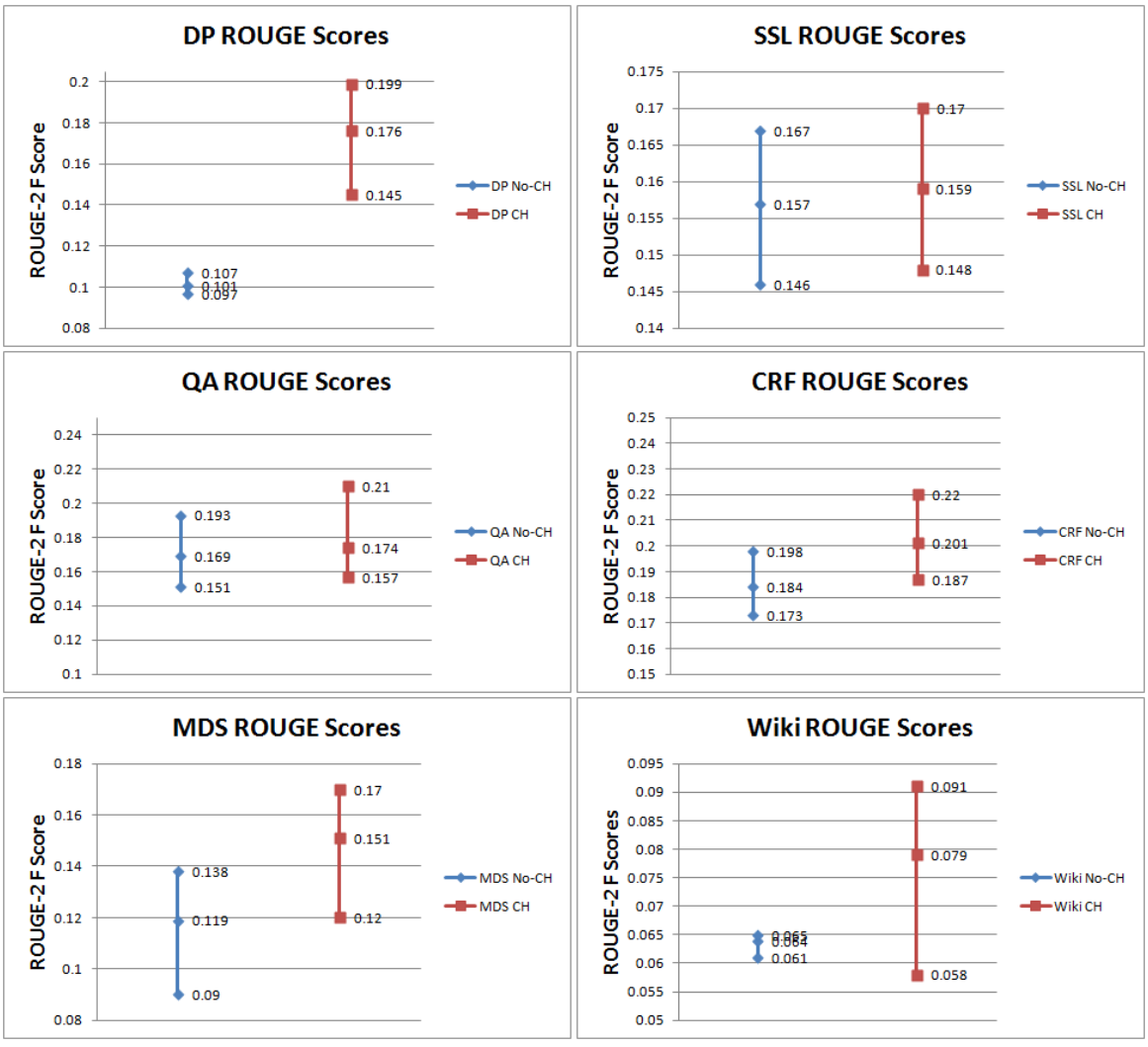

Figure 6.3: ROUGE-2 Scores with 95% confidence intervals for Dependency Parsing (DP), Question Answering (QA), Multi-document Summarization (MDS), Semi-supervised Learning (SSL), Conditional Random Fields (CRF), and Wikipedia (Wiki).

Human Performance: ROUGE-2					
Dataset	Hum1	Hum2	Hum3	Hum4	Avg
QA	0.1807	0.1956	0.0756	0.2019	0.1635
DP	0.1550	0.1259	0.1200	0.1654	0.1416

Table 6.2: ROUGE-2 scores of human-created summaries of QA and DP data. ROUGE-1 and ROUGE-L followed similar patterns.

Dataset	Hum1	Hum2	Hum3	Hum4	Hum5	Avg
CRF	0.241	0.205	0.229	0.249	N/A	0.231
SSL	0.181	0.247	0.172	0.243	0.201	0.214
MDS	0.205	0.195	0.201	0.190	N/A	0.198
wiki	0.161	0.181	0.184	0.177	N/A	0.176

Table 6.3: ROUGE-2 scores of human-created summaries of the Conditional Random Fields (CRF), Semi-supervised Learning (SSL), Multi-document Summarization (MDS), and Wikipedia (wiki) data sets.

System Performance: ROUGE-2			
Dataset	Random	MASCS	MASCS-CH
QA	0.116	0.169	**0.173**
DP	0.107	0.101	**0.139**
CRF	0.111	0.184	**0.201**
SSL	0.150	0.157	**0.159**
MDS	0.102	0.119	**0.151**
wiki	0.058	0.064	**0.079**

Table 6.4: ROUGE-2 F-measure scores of automatic summaries of all the Question Answering (QA), Dependency Parsing (DP), Conditional Random Fields (CRF), Semi-supervised Learning (SSL), Multi-document Summarization (MDS), and Wikipedia (wiki) data sets. MASCS is the original MASCS system without citation handling; MASCS-CH is the version of MASCS with citation handling.

6.4.3 Pyramid

For my second approach to evaluation on MASCS, I used a nugget-based evaluation methodology. Three impartial annotators (knowledgeable in NLP but not affiliated with the project) reviewed the citation texts and/or abstract sets for each of the papers in the QA and DP sets and manually extracted prioritized lists of 2–8 "nuggets," or main contributions, supplied by each paper. Each nugget was assigned a weight based on the frequency with which it was listed by annotators as well as the priority it was assigned in each case. The automatically generated summaries from MASCS were then scored based on the number and weight of the nuggets that they covered. This evaluation approach is similar to the one adopted by Qazvinian and Radev (2008), but adapted here for use in the multi-document case.

The annotators were instructed to extract nuggets for each of the 10 QA and 16 DP papers, based only on the citation texts for those papers. The weight for each nugget was obtained by reversing its priority out of 8 (e.g., a nugget listed with priority 1 was assigned a weight of 8) and summing the weights over each listing of that nugget.[2]

To evaluate a given summary, I counted the number and weight of nuggets that it covered. Nuggets were detected via the combined use of annotator-provided regular expressions and careful human review. Recall was calculated by dividing the combined weight of covered nuggets by the combined weight of all nuggets in

[2]Results obtained with other weighting schemes that ignored priority ratings and multiple mentions of a nugget by a single annotator showed the same trends as the ones shown by the selected weighting scheme, but the latter was a stronger distinguisher among the evaluated systems.

Human Performance: Pyramid F-measure					
Input	Hum1	Hum2	Hum3	Hum4	Avg
QA	0.350	0.458	0.403	0.577	0.447
DP	0.179	0.467	0.362	0.513	0.380

Table 6.5: Pyramid F-measure scores of human-created summaries of QA and DP data.

System Performance: Pyramid F-measure			
Input	Random	MASCS	MASCS-CH
QA	0.321	**0.422**	0.410
DP	0.219	0.241	**0.298**

Table 6.6: Pyramid F-measure scores of automatic summaries of QA and DP data. The summaries are evaluated using nuggets drawn from QA and DB citation texts. MASCS is the original MASCS system without citation handling; MASCS-CH is the version of MASCS with citation handling.

the nugget set. Precision was calculated by dividing the number of distinct nuggets covered in a summary by the number of sentences constituting that summary, with a cap of 1. F-measure, the weighted harmonic mean of precision and recall, was calculated with a beta value of 3 in order to assign the greatest weight to recall. Recall is favored because it rewards summaries that include highly weighted (important) facts, rather than just a great number of facts.

Table 6.5 gives the F-measure values of the 250-word summaries manually generated by humans. The summaries were evaluated using the nuggets drawn from the QA citation texts, QA abstracts, and DP citation texts. The average of their scores (listed in the rightmost column) may be considered a good score to aim for by the automatic summarization methods.

Table 6.6 gives the F-measure values of the surveys generated by the random summarizer and the two variants of MASCS, evaluated using nuggets drawn from the QA and DP citation texts. Among the various automatic summarizers, neither

MASCS or MASCS-CH performed significantly better than the other at this task.

This chapter has examined the effects of citation handling on three NLP components. For the Stanford Parser, citation handling provides better confidence scores and a 3.73x improvement in run time. For Trimmer, perplexity scores from a language model for sentence compression candidates are better when using citation handling. For summarization, the ROUGE scores for summaries generated by MASCS with citation handling are significantly higher than those without citation handling. However, the Pyramid scores for MASCS with and without citation handling are relatively comparable. This can be attributed to the fact that Pyramid measures whether certain "nuggets" of information are included in a summary, and is not a judgment on fluency (additional judgments of fluency are presented as future work). Despite this, citation handling has been shown to significantly improve the quality and performance of the Stanford Parser, Trimmer, and MASCS.

Chapter 7

Conclusion and Future Work

In this thesis, I have presented issues that arise with parsers and summarization systems on documents containing citations in scientific literature. I identified two different types of citations, constituent and parenthetical citations, based on their syntactic properties in a citation sentence. An approach to preprocessing these citations, called citation handling, was presented as a solution to the issues with parsers and summarization systems. With citation handling, constituent citations are replaced with filler text containing a unique identifier, and parenthetical citations are removed. I have also provided a means for re-inserting both these types of citations back into a citation sentence: the unique identifier used to replace constituent citations allows for easy re-insertion of the original citations, and parenthetical citations can be re-inserted at the end of the citation sentence.

The effects of citation handling on three NLP components (the Stanford Parser, Trimmer, and MASCS) was investigated, and several specific examples were provided for each component demonstrating the positive impact of citation handling. In addition, several standard evaluations were performed on each of the components, and citation handling was shown to have a significant effect in each case. As a result, this thesis has shown that citation handling can improve NLP components dealing with scientific literature, and any NLP system that relies on parse trees can benefit

from using citation handling.

Future work includes implementing a better means of re-inserting parenthetical citations back in to the sentence candidates. As stated before, it is difficult to determine what part of a parse tree a parenthetical citation is associated with, and therefore it is hard to decide whether to re-insert a parenthetical citation after sentence compression. Currently, the citations are appended to the end of the sentence rather than in their original location in the sentence. An investigation into the Stanford Parser confidence scores is also needed, or an alternative avenue for evaluating the parse trees could be used.

Expanding the analysis from just citations to all parenthetical phrases is another area for future work. Some examples of other parenthetical phrases can include abbreviations and sidenotes to supplement the "main" sentence. Since using citation handling results in generating parse trees almost four times faster than not using citation handling, it seems to suggest that parenthetical phrases trip up parsers, and expanding this investigation could prove helpful. In addition, expanding the analysis to determine the effect on other NLP components besides the three examined in this thesis is another avenue for future work. Other areas such as sentiment analysis use parse trees as part of their processes, and it would be interesting to see the impact of the better parse trees from citation handling on these other NLP components.

Another avenue of future research is to carry out additional Turk tasks to determine the effectiveness of citation handling in generating fluent summaries. As was pointed out in Section 6.4.3, an additional judgment on fluency could prove

beneficial for evaluating summarization, since citation handling performed better with respect to ROUGE but relatively the same with Pyramid. Mechanical Turk Tasks would be created where Turkers would judge the fluency for summaries that used citation handling, ones that did not use citation handling, and summaries generated using bag of words. Ultimately, I intend to apply the techniques described herein for the purpose of providing summaries of topically organized technical and scientific texts.

Finally, citation handling will be used as a component of a larger system that discovers patterns of emergence and connections between technical concepts within full-text scientific, technical, and patent literatures.

Bibliography

Amjad Abu-Jbara and Dragomir Radev. Coherent citation-based summarization of scientific papers. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 500–509, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

Shannon Bradshaw. *Reference Directed Indexing: Indexing Scientific Literature in the Context of Its Use.* PhD thesis, Northwestern University, 2002.

Shannon Bradshaw. Reference directed indexing: Redeeming relevance for subject search in citation indexes. In *Proceedings of the 7th European Conference on Research and Advanced Technology for Digital Libraries*, 2003.

Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '98, pages 335–336, New York, NY, USA, 1998. ACM. ISBN 1-58113-015-5. doi: 10.1145/290941.291025. URL `http://doi.acm.org/10.1145/290941.291025`.

Hal Daumé III. Megam: Maximum entropy model optimization package. ACL Data and Code Repository, ADCR2008C003, http://aclweb.org/aclwiki, 2008.

Aaron Elkiss, Siwei Shen, Anthony Fader, Güneş Erkan, David States, and Dragomir R. Radev. Blind men and elephants: What do citation summaries tell us about a research article? *Journal of the American Society for Information Science and Technology*, 59(1):51–62, 2008.

Robert Gove, Cody Dunne, Ben Shneiderman, Judith Klavans, and Bonnie Dorr. Evaluating visual and statistical exploring of scientific literature networks. In *VL/HCC'11*, 2011.

Wesley Hildebrandt, Boris Katz, and Jimmy Lin. Overview of the trec 2003 question-answering track. In *Proceedings of the 2004 Human Language Technology Conference and the North American Chapter of the Association for Computational Linguistics Annual Meeting (HLT/NAACL 2004)*, 2004.

Mark Joseph and Dragomir Radev. Citation analysis, centrality, and the ACL Anthology. Technical Report CSE-TR-535-07, University of Michigan. Dept.of Electrical Engineering and Computer Science, 2007.

Dan Klein and Christopher D. Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Meeting of ACL*, pages 423–430, 2003.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Proceedings of the ACL workshop on Text Summarization Branches Out*, 2004.

Jimmy J. Lin and Dina Demner-Fushman. Methods for automatically evaluating answers to complex questions. *Information Retrieval*, 9(5):565–587, 2006.

Qiaozhu Mei and ChengXiang Zhai. Generating impact-based summaries for scientific literature. In *Proceedings of ACL '08*, pages 816–824, 2008.

Saif Mohammad, Bonnie Dorr, Melissa Egan, Ahmed Hassan, Pradeep Muthukrishan, Vahed Qazvinian, Dragomir Radev, and David Zajic. Using citations to generate surveys of scientific paradigms. In *Proceedings of NAACL-HLT 2009*, 2009.

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

Ani Nenkova and Rebecca Passonneau. Evaluating content selection in summarization: The pyramid method. *Proceedings of the HLT-NAACL conference*, 2004.

Rebecca Passonneau, Nizar Habash, and Owen Rambow. Inter-annotator agreement on a multilingual semantic annotation task. In *In Proceedings of LREC*, 2006.

Vahed Qazvinian and Dragomir R. Radev. Scientific paper summarization using citation summary networks. In *COLING 2008*, Manchester, UK, 2008.

Andreas Stolcke. SRILM – an extensible language modeling toolkit. In *Proceedings of ICSLP*, volume 2, pages 901–904, Denver, USA, 2002.

Simone Teufel, Advaith Siddharthan, and Dan Tidhar. Automatic classification of citation function. In *Proceedings of EMNLP*, pages 103–110, Australia, 2006.

Ellen M. Voorhees. Overview of the trec 2003 question answering track. In *Proceedings of the Twelfth Text Retrieval Conference (TREC 2003)*, 2003.

David M. Zajic, Bonnie J. Dorr, Jimmy Lin, and Richard Schwartz. Multi-candidate reduction: Sentence compression as a tool for document summarization tasks. *Information Processing and Management (Special Issue on Summarization)*, 2007.