Reference Scope Identification in Citing Sentences

Amjad Abu-Jbara
EECS Department
University of Michigan
Ann Arbor, MI, USA
amjbara@umich.edu

Dragomir Radev
EECS Department
University of Michigan
Ann Arbor, MI, USA
radev@umich.edu

Abstract

A citing sentence is one that appears in a scientific article and cites previous work. Citing sentences have been studied and used in many applications. For example, they have been used in scientific paper summarization, automatic survey generation, paraphrase identification, and citation function classification. Citing sentences that cite multiple papers are common in scientific writing. This observation should be taken into consideration when using citing sentences in applications. For instance, when a citing sentence is used in a summary of a scientific paper, only the fragments of the sentence that are relevant to the summarized paper should be included in the summary. In this paper, we present and compare three different approaches for identifying the fragments of a citing sentence that are related to a given target reference. Our methods are: word classification, sequence labeling, and segment classification. Our experiments show that segment classification achieves the best results.

1 Introduction

Citation plays an important role in science. It makes the accumulation of knowledge possible. When a reference appears in a scientific article, it is usually accompanied by a span of text that highlights the important contributions of the cited article. We call a sentence that contains an explicit reference to previous work a *citation sentence*. For example, sentence (1) below is a citing sentence that cites a paper by Philip Resnik and describes the problem Resnik addressed in his paper.

*(1) **Resnik (1999)** addressed the issue of language identification for finding Web pages in the languages of interest.*

Previous work has studied and used citation sentences in various applications such as: scientific paper summarization (Elkiss et al., 2008; Qazvinian and Radev, 2008; Mei and Zhai, 2008; Qazvinian et al., 2010; Qazvinian and Radev, 2010; Abu-Jbara and Radev, 2011), automatic survey generation (Nanba et al., 2000; Mohammad et al., 2009), citation function classification (Nanba et al., 2000; Teufel et al., 2006; Siddharthan and Teufel, 2007; Teufel, 2007), and paraphrase recognition (Nakov et al., 2004; Schwartz et al., 2007).

Sentence (1) above contains one reference, and the whole sentence is talking about that reference. This is not always the case in scientific writing. Sentences that contain references to multiple papers are very common. For example, sentence (2) below contains three references.

*(2) Grefenstette and Nioche (2000) and Jones and Ghani (2000) use the web to generate corpora for languages where electronic resources are scarce, while **Resnik (1999)** describes a method for mining the web for bilingual texts.*

2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 80–90,
Montréal, Canada, June 3-8, 2012. ©2012 Association for Computational Linguistics

The first fragment describes the contribution of Grefenstette and Nioche (2000) and Jones and Ghani (2000). The second fragment describes the contribution of Resnik (1999).

This observation should be taken into consideration when using citing sentences in the aforementioned applications. For example, in citation-based summarization of scientific papers, a subset of citing sentences that cite a given target paper is selected and used to form a summary of that paper. It is very likely that one or more of the selected sentences cite multiple papers besides the target. This means that some of the text included in the summary might be irrelevant to the summarized paper. Including irrelevant text in the summary introduces several problems. First, the summarization task aims at summarizing the contributions of the target paper using minimal text. Extraneous text takes space in the summary while being irrelevant and less important. Second, including irrelevant text in the summary breaks the context and confuses the reader. Therefore, if sentence (2) above is to be added to a citation-based summary of Resnikś (1999) paper, only the underlined fragment should be added to the summary and the rest of the sentence should be excluded.

For another example, consider the task of citation function classification. The goal of this task is to determine the reason for citing paper B by paper A based on linguistic and structural features extracted from citing sentences that appear in A and cite B. If a citing sentence in A cites multiple papers besides B, classification features should be extracted only from the fragments of the sentence that are relevant to B. Sentence (3) below shows an examples of this case.

(3) Cohn and Lapata (2008) used the GHKM extraction method (Galley et al., 2004), which is limited to constituent phrases and thus produces a reasonably small set of syntactic rules.

If the target reference is Cohn and Lapata (2008), only the underlined segment should be used for feature extraction. The limitation stated in the second segment of sentence is referring to Galley et al., (2004).

In this paper, we address the problem of identifying the fragments of a citing sentence that are related to a given target reference. Henceforth, we use the term *Reference Scope* to refer to those fragments. We present and compare three different approaches to this problem.

In the first approach, we define the problem as a word classification task. We classify each word in the sentence as *inside* or *outside* the scope of the target reference.

In the second approach, we define the problem as a sequence labeling problem. This is different from the first approach in that the label assigned to each word is dependent on the labels of nearby words. In the third approach, instead of classifying individual words, we split the sentence into segments and classify each segment as *inside* or *outside* the scope of the target reference.

Applying any of the three approaches is preceded by a preprocessing stage. In this stage, citing sentences are analyzed to tag references, identify groups of references, and distinguish between syntactic and non-syntactic references.

The rest of this paper is organized as follows. Section 2 examines the related work. We define the problem in Section3. Section 4 presents our approaches. Experiments, results and analysis are presented in Section 5. We conclude and provide directions to future work in Section 6

2 Related Work

Our work is related to a large body of research on citations (Hodges, 1972; Garfield et al., 1984). The interest in studying citations stems from the fact that bibliometric measures are commonly used to estimate the impact of a researcher's work (Borgman and Furner, 2002; Luukkonen, 1992). White (2004) provides a good recent survey of the different research lines that use citations. In this section we review the research lines that are relevant to our work and show how our work is different.

One line of research that is related to our work has to do with identifying what Nanba and Okumura (1999) call the *citing area* They define the citing area as the succession of sentences that appear around the location of a given reference in a scientific paper and have connection to it. Their algorithm starts by adding the sentence that contains the target reference as the first member sentence in the citing area. Then, they use a set of cue words and hand-crafted rules to determine whether the surrounding sentences should be added to the citing area or not. In (Nanba et al., 2000) they use their citing area identification algorithm to improve citation type classification and automatic survey generation.

Qazvinian and Radev (2010) addressed a similar problem. They proposed a method based on probabilistic inference to extract non-explicit citing sentences; i.e., sentences that appear around the sentence that contains the target reference and are related to it. They showed experimentally that citation-based survey generation produces better results when using both explicit and non-explicit citing sentences rather than using the explicit ones alone.

Although this work shares the same general goal with ours (i.e identifying the pieces of text that are relevant to a given target reference), our work is different in two ways. First, previous work mostly ignored the fact that the citing sentence itself might be citing multiple references. Second, it defined the *citing area* (Nanba and Okumura, 1999) or the *citation context* (Qazvinian and Radev, 2010) as a set of whole contiguous sentences. In our work, we address the case where one citing sentence cites multiple papers, and define what we call the *reference scope* to be the fragments (not necessarily contiguous) of the citing sentence that are related to the target reference.

In a recent work on citation-based summarization by Abu-Jbara and Radev (2011), the authors noticed the issue of having multiple references in one sentence. They raised this issue when they discussed the factors that impede the coherence and the readability of citation-based summaries. They suggested that removing the fragments of a citing sentence that are not relevant to the summarized paper will significantly improve the quality of the produced summaries. In their work, they defined the scope of a given reference as the shortest fragment of the citing sentence that contains the reference and could form a grammatical sentence if the rest of the sentence was removed. They identify the scope by generating the syntactic parse tree of the sentence and then finding the text that corresponds to the smallest subtree rooted at an S node and contains the target reference node as one of its leaf nodes. They admitted that their method was very basic and works only when the scope forms one grammatical fragment, which is not true in many cases.

Athar (2011) noticed the same issue with citing sentences that cite multiple references, but this time in the context of sentiment analysis in citations. He showed experimentally that identifying what he termed the *scope of citation influence* improves sentiment classification accuracy. He adapted the same basic method proposed by Abu-Jbara and Radev (2011). We use this method as a baseline in our evaluation below.

In addition to this related work, there is a large body of research that used citing sentences in different applications. For example, citing sentences have been used to summarize the contributions of a scientific paper (Qazvinian and Radev, 2008; Qazvinian et al., 2010; Qazvinian and Radev, 2010; Abu-Jbara and Radev, 2011). They have been also used to generate surveys of scientific paradigms (Nanba and Okumura, 1999; Mohammad et al., 2009). Several other papers analyzed citing sentences to recognize the citation function; i.e., the author's reason for citing a given paper (Nanba et al., 2000; Teufel et al., 2006; Teufel, 2007). Schwartz et al. (2007) proposed a method for aligning the words within citing sentences that cite the same paper. The goal of his work was to aid named entity recognition and paraphrase identification in scientific papers.

We believe that all the these applications will benefit from the output of our work.

3 Problem Definition

The problem that we are trying to solve is to identify which fragments of a given citing sentence that cites multiple references are semantically related to a given target reference. As stated above, we call these fragments the *reference scope*. Formally, given a citing sentence $S = \{w1, w2, ..., w_n\}$ where $w1, w2, ..., w_n$ are the tokens of the sentence and given that S contains a set of two or more references R, we want to assign the label 1 to the word w_i if it falls in the scope of a given target reference $r \in R$, and 0 otherwise.

For example, sentences (4) and (5) below are labeled for the target references Tetreault and Chodorow (2008), and Cutting et al.(1992) respectively. The underlined words are labeled 1 (i.e., inside the target reference scope), while all others are labeled 0.

(4) For example, ***Tetreault and Chodorow (2008)*** *use a maximum entropy classifier to build a model of correct preposition usage, with 7 million instances in their training set, and Lee and Knutsson (2008) use memory-based learning, with 10 million sentences in their training set.*

*(5) There are many POS taggers developed using different techniques for many major languages such as transformation-based error-driven learning (Brill, 1995), decision trees (Black et al., 1992), Markov model (***Cutting et al., 1992***), maximum entropy methods (Ratnaparkhi, 1996) etc for English.*

4 Approach

In this section, we present our approach for addressing the problem defined in the previous section. Our approach involves two stages: 1) preprocessing and 2) reference scope identification. We present three alternative methods for the second stage. The following two subsections describe the two stages.

4.1 Stage 1: Preprocessing

The goal of the preprocessing stage is to clean and prepare the citing sentence for the next processing steps. The second stage involves higher level text processing such as part-of-speech tagging, syntactic parsing, and dependency parsing. The available tools for these tasks are not trained on citing sentences which contain references written in a special format. For example, it is very common in scientific writing to have references (usually written between parentheses) that are not a syntactic part of the sentence. It is also common to cite a group of references who share the same contribution by listing them between parentheses separated by a comma or a semi-colon. We address these issues to improve the accuracy of the processing done in the second stage. The preprocessing stage involves three tasks:

4.1.1 Reference Tagging

The first preprocessing task is to find and tag all the references that appear in the citing sentence. Authors of scientific articles use standard patterns to include references in text. We apply a regular expression to find all the references that appear in a sentence. We replace each reference with a placeholder. The target reference is replaced by TREF. Each other reference is replaced by REF. We keep track of the original text of each replaced reference. Sentence (6) below shows an example of a citing sentence with the references replaced.

(6) These constraints can be lexicalized (REF.1; REF.2), unlexicalized (REF.3; ***TREF.4****) or automatically learned (REF.5; REF.6).*

4.1.2 Reference Grouping

It is common in scientific writing to attribute one contribution to a group of references. Sentence (6) above contains three groups of references. Each group constitutes one entity. Therefore, we replace each group with a placeholder. We use GTREF to replace a group of references that contains the target reference, and GREF to replace a group of references that does *not* contain the target reference.

Sentence (7) below is the same as sentence (6) but with the three groups of references replaced.

*(7) These constraints can be lexicalized (GREF.1), unlexicalized (**GTREF.2**) or automatically learned (GREF.3).*

4.1.3 Non-syntactic Reference Removal

A reference (REF or TREF) or a group of references (GREF or GTREF) could either be a syntactic constituent and has a semantic role in the sentence (e.g. GTREF.1 in sentence (8) below) or not (e.g. REF.2 in sentence (8)).

(8) (GTREF.1) apply fuzzy techniques for integrating source syntax into hierarchical phrase-based systems (REF.2).

The task in this step is to determine whether a reference is a syntactic component in the sentence or not. If yes, we keep it as is. If not, we remove it from the sentence and keep track of its position. Accordingly, after this step, REF.2 in sentence (8) will be removed. We use a rule-based algorithm to determine whether a reference should be removed from the sentence or kept. Our algorithm (Algorithm 1) uses stylistic and linguistic features such as the style of the reference, the position of the reference, and the surrounding words to make the decision.

When a reference is removed, we pick a word from the sentence to represent it. This is needed for feature extraction in the next stage. We use as a representative the head of the closest noun phrase (NP) that comes before the position of the removed reference. For example, in sentence (8) above, the closest NP before REF.2 is *hierarchical phrase-based systems* and the head is the noun *systems*.

4.2 Stage 2: Reference Scope Identification

In this section we present three different methods for identifying the scope of a given reference within a citing sentence. We compare the performance of these methods in Section 5. The following three subsections describe the methods.

Algorithm 1 Remove Non-syntactic References

```
Require: A citing sentence S
for all Reference R (REF, TREF, GREF, or GTREF) in S do
    if R style matches "Authors (year)" then
        Keep R // syntactic
    else if R is the first word in the sentence or in a clause then
        Keep R // syntactic
    else if R is preceded by a preposition (in, of, by, etc.) then
        Keep R // syntactic
    else
        Remove R // non-syntactic
    end if
end for
```

4.2.1 Word Classification

In this method we define reference scope identification as a classification task of the individual words of the citing sentence. Each word is classified as *inside* or *outside* the scope of a given target reference. We use a number of linguistic and structural features to train a classification model on a set of labeled sentences. The trained model is then used to label new sentences. The features that we use to train the model are listed in Table 1. We use the Stanford parser (Klein and Manning, 2003) for syntactic and dependency parsing. We experiment with two classification algorithms: Support Vector Machines (SVM) and logistic regression.

4.2.2 Sequence Labeling

In the method described in Section 4.2.1 above, we classify each word independently from the labels of the nearby words. The nature of our task, however, suggests that the accuracy of word classification can be improved by considering the labels of the words surrounding the word being classified. It is very likely that the word takes the same label as the word before and after it if they all belong to the same clause in the sentence. In this method we define the problem as a sequence labeling task. Now, instead of looking for the best label for each word individually, we look for the globally best sequence

Feature	Description
Distance	The distance (in words) between the word and the target reference.
Position	This feature takes the value 1 if the word comes before the target reference, and 0 otherwise.
Segment	After splitting the sentence into segments by punctuation and coordination conjunctions, this feature takes the value 1 if the word occurs in the same segment with the target reference, and 0 otherwise.
Part of speech tag	The part of speech tag of the word, the word before, and the word after.
Dependency Distance	Length of the shortest dependency path (in the dependency parse tree) that connects the word to the target reference or its representative. It has been shown in previous work on relation extraction that the shortest path between any two entities captures the information required to assert a relationship between them (Bunescu and Mooney, 2005)
Dependency Relations	This item includes a set of features. Each features corresponds to a dependency relation type. If the relation appears in the dependency path that connects the word to the target reference or its representative, its corresponding feature takes the value 1, and 0 otherwise.
Common Ancestor Node	The type of the node in the syntactic parse tree that is the least common ancestor of the word and the target reference.
Syntactic Distance	The number of edges in the shortest path that connects the word and the target reference in the syntactic parse tree.

Table 1: The features used for word classification and sequence labeling

of labels for all the words in the sentence at once.

We use Conditional Random Fields (CRF) as our sequence labeling algorithm. In particular, we use first-order chain-structured CRF. The chain consists of two sets of nodes: a set of hidden nodes **Y** which represent the scope labels (0 or 1) in our case, and a set of observed nodes **X** which represent the observed features. The task is to estimate the probability of a sequence of labels Y given the sequence of observed features X: $P(\mathbf{Y}|\mathbf{X})$

Lafferty et al. (2001) define this probability to be a normalized product of potential functions ψ:

$$P(\mathbf{y}|\mathbf{x}) = \prod_t \psi_k(y_t, y_{t-1}, \mathbf{x}) \tag{1}$$

Where $\psi_k(y_t, y_{t-1}, \mathbf{x})$ is defined as

$$\psi_k(y_t, y_{t-1}, \mathbf{x}) = exp(\sum_k \lambda_k f(y_t, y_{t-1}, \mathbf{x})) \tag{2}$$

where $f(y_t, y_{t-1}, \mathbf{x})$ is a transition feature function of the label at positions $i - 1$ and i and the observation sequence **x**; and λ_j is parameter to be estimated from training data. We use, as the observations at each position, the same features that we used in Section 4.2.1 above (Table 1).

4.2.3 Segment Classification

We noticed that the scope of a given reference often consists of units of higher granularity than words. Therefore, in this method, we split the sentence into segments of contiguous words and, instead of labeling individual words, we label the whole segment as *inside* or *outside* the scope of the target reference. We experimented with two different segmentation methods. In the first method (method-1), we segment the sentence at punctuation marks, coordination conjunctions, and a set of special expressions such as “for example”, “for instance”, “including”, “includes”, “such as”, “like”, etc. Sentence (8) below shows an example of this segmentation method (Segments are enclosed in square brackets).

(8) [Rerankers have been successfully applied to numerous NLP tasks such as] [parse selection (GTREF)], [parse reranking (GREF)], [question-answering (REF)].

In the second segmentation method (method-2), we split the sentence into segments of finer granularity. We use a chunking tool to identify noun groups, verb groups, preposition groups, adjective groups, and adverb groups. Each such group (or chunk) forms a segment. If a word does not belong to any chunk, it forms a singleton segment by itself. Sentence (9) below shows an example of this segmentation method (Segments are enclosed in square brackets).

(9) [To] [score] [the output] [of] [the coreference models], [we] [employ] [the commonly-used MUC scoring program (REF)] [and] [the recently-developed CEAF scoring program (TREF)].

We assign a label to each segment in two steps. In the first step, we use the sequence labeling method described in Section 4.2.2 to assign labels to all the individual words in the sentence. In the second step, we aggregate the labels of all the words contained in a segment to assign a label to the whole segment. We experimented with three different label aggregation rules: 1) rule-1: assign to the segment the majority label of the words it contains, and 2) rule-2: assign to the segment the label 1 (i.e., *inside*) if at least one of the words contained in the segment is labeled 1, and assign the label 0 to the segment otherwise, and 3) rule-3: assign the label 0 to the segment if at least of the words it contains is labeled 0, and assign 1 otherwise.

5 Evaluation

5.1 Data

We use the ACL Anthology Network corpus (AAN) (Radev et al., 2009) in our evaluation. AAN is a publicly available collection of more than 19,000 NLP papers. AAN provides a manually curated citation network of its papers and the citing sentence(s) associated with each edge. The current release of AAN contains about 76,000 unique citing sentences 56% of which contain 2 or more references and 44% contain 1 reference only. From this set, we randomly selected 3500 citing sentences, each containing at least two references (3.75 references on average with a standard deviation of 2.5). The total number of references in this set of sentences is 19,591.

We split the data set into two random subsets: a development set (200 sentences) and a training/testing set (3300 sentences). We used the development set to study the data and develop our strategies of addressing the problem. The second set was used to train and test the system in a cross-validation mode.

5.2 Annotation

We asked graduate students with good background in NLP (the area of the annotated sentences) to provide three annotations for each sentence in the data set described above. First, we asked them to determine whether each of the references in the sentence was correctly tagged or not. Second, we asked them to determine for each reference whether it is a syntactic constituent in the sentence or not. Third, we asked them to determine and label the scope of one reference in each sentence which was marked as a target reference (TREF). We designed a user-friendly tool to collect the annotations from the students.

To estimate the inter-annotator agreement, we picked 500 random sentences from our data set and assigned them to two different annotators. The inter-annotator agreement was perfect on both the reference tagging annotation and the reference syntacticality annotation. This is expected since both are objective, clear, and easy tasks. To measure the inter-annotator agreement on the scope annotation task, we deal with it as a word classification task. This allows us to use the popular classification agreement measure, the Kappa coefficient (Cohen, 1968). The Kappa coefficient is defined as follows:

$$K = \frac{P(A) - P(E)}{1 - P(E)} \tag{3}$$

where P(A) is the relative observed agreement among raters and P(E) is the hypothetical probability of chance agreement. The agreement between the two annotators on the scope identification task was $K = 0.61$. On Landis and Kochs (Landis and Koch, 1977) scale, this value indicates substantial agreement.

5.3 Experimental Setup

We use the Edinburgh Language Technology Text Tokenization Toolkit (LT-TTT) (Grover et al., 2000) for text tokenization, part-of-speech tagging, chunking, and noun phrase head identification. We use the Stanford parser (Klein and Manning, 2003) for syntactic and dependency parsing. We use LibSVM (Chang and Lin, 2011) for Support Vector Machines (SVM) classification. Our SVM model uses a linear kernel. We use Weka (Hall et al., 2009) for logistic regression classification. We use the Machine Learning for Language Toolkit (MALLET) (McCallum, 2002) for CRF-based sequence labeling. In all the scope identification experiments and results below, we use 10-fold cross validation for training/testing.

5.4 Preprocessing Component Evaluation

We ran our three rule-based preprocessing modules on the testing data set and compared the output to the human annotations. The test set was not used in the tuning of the system but was done using the development data set as described above. We report the results for each of the preprocessing modules. Our reference tagging module achieved 98.3% precision and 93.1% recall. Most of the errors were due to issues with text extraction from PDF or due to bad references practices by some authors (i.e., not following scientific referencing standards). Our reference grouping module achieved perfect accuracy for all the correctly tagged references. This was expected since this is a straightforward task. The non-syntactic reference removal module achieved 90.08% precision and 90.1% recall. Again, most of the errors were the result of bad referencing practices by the authors.

5.5 Reference Scope Identification Experiments

We conducted several experiments to compare the methods proposed in Section 4 and their variants. We ran all the experiments on the training/testing set (the 3300 sentences) described in Section 5.1. The experiments that we ran are as follows: 1) word classification using a SVM classifier (WC-SVM); 2) word classification using a logistic regression classifier(WC-LR); 3) CRF-based sequence labeling (SL-CRF); 4) segment classification using segmentation method-1 and label aggregation rule-1 (SC-S1-R1); 5,6,7,8,9) same as (4) but using different combinations of segmentation methods 1 and 2, and label aggregation rules 1,2 and 3: SC-S1-R2, SC-S1-R3, SC-S2-R1, SC-S2-R2, SC-S2-R3 (where Sx refers to segmentation method x and Ry refers to label aggregation rule y all as explained in Section 4.2.3). Finally, 10) we compare our methods to the baseline method proposed by Abu-Jbara and Radev (2011) which was described in Section 4 (AR-2011).

Method	Accuracy	Precision	Recall	F-measure
AR-2011	54.0%	63.3%	33.1%	41.5%
WC-SVM	74.9%	74.5%	93.4%	82.9%
WC-LR	74.3%	76.8%	88.0%	82.0%
SL-CRF	78.2%	80.1%	94.2%	86.6%
SC-S1-R1	73.7%	72.1%	97.8%	83.0%
SC-S1-R2	69.3%	68.4%	**98.9%**	80.8%
SC-S1-R3	60.0%	61.8%	73.3%	60.9%
SC-S2-R1	**81.8%**	**81.2%**	93.8%	**87.0%**
SC-S2-R2	78.2%	77.3%	94.9%	85.2%
SC-S2-R3	66.1%	67.1%	71.2%	69.1%

Table 3: Results of scope identification using the different algorithms described in the paper

To better understand which of the features listed in Table 1 are more important for the task, we use Guyon et al.'s (2002) method for feature selection using SVM to rank the features based on their importance. The results of the experiments and the feature analysis are presented and discussed in the following subsection.

5.6 Results and Discussion

5.6.1 Experimental Results

We ran the experiments described in the previous subsection on the testing data described in Sec-

	Method	Output
Example 1	Word Classification (WC-SVM)	A wide range of contextual information, such as surrounding words (GREF), dependency or case structure (GTREF), and dependency path (GREF), has been utilized for similarity calculation, and achieved considerable success.
	Sequence Labeling (SL-CRF)	A wide range of contextual information, such as surrounding words (GREF), dependency or case structure (GTREF), and dependency path (GREF), has been utilized for similarity calculation, and achieved considerable success.
	Segment Classification (SC-S2-R1)	A wide range of contextual information, such as surrounding words (GREF), dependency or case structure (GTREF), and dependency path (GREF), has been utilized for similarity calculation, and achieved considerable success.
Example 2	Word Classification (WC-SVM)	Some approaches have used WordNet for the generalization step (GTREF), others EM-based clustering (REF).
	Sequence Labeling (SL-CRF)	Some approaches have used WordNet for the generalization step (GTREF), others EM-based clustering (REF).
	Segment Classification (SC-S2-R1)	Some approaches have used WordNet for the generalization step (GTREF), others EM-based clustering (REF).

Table 2: Two example outputs produced by the three methods

tion 5.1. Table 3 compares the precision, recall, F1, and accuracy for the three methods described in Section 4 and their variations. All the metrics were computed at the word level. The results show that all our methods outperform the baseline method AR-2011 that was proposed by Abu-Jbara and Radev (2011). In the word classification method, we notice no significant difference between the performance of the SVM vs Logistic Regression classifier. We also notice that the CRF-based sequence labeling method performs significantly better than the word classification method. This result corroborates our intuition that the labels of neighboring words are dependent. The results also show that segment labeling generally performs better than word labeling. More specifically, the results indicate that segmentation based on chunking and the label aggregation based on plurality when used together (i.e., SC-S2-R1) achieve higher precision, accuracy, and F-measure than the punctuation-based segmentation and the other label aggregation rules.

Table 2 shows the output of the three methods on two example sentences. The underlined words are labeled by the system as scope words.

5.6.2 Feature Analysis

We performed an analysis of our classification features using Guyon et al. (2002) method. The analysis revealed that both structural and syntactic features are important. Among the syntactic features, the dependency path is the most important. Among the structural features, the *segment* feature (as described in Table 1) is the most important.

6 Conclusions

We presented and compared three different methods for reference scope identification: word classification, sequence labeling, and segment classification. Our results indicate that segment classification achieves the best performance. The next direction in this research is to extract the scope of a given reference as a standalone grammatical sentence. In many cases, the scope identified by our method can form a grammatical sentence with no or minimal postprocessing. In other cases, more advanced *text regeneration* techniques are needed for scope extraction.

References

Amjad Abu-Jbara and Dragomir Radev. 2011. Coherent citation-based summarization of scientific papers. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Lan-*

guage Technologies, pages 500–509, Portland, Oregon, USA, June. Association for Computational Linguistics.

Awais Athar. 2011. Sentiment analysis of citations using sentence structure-based features. In *Proceedings of the ACL 2011 Student Session*, pages 81–87, Portland, OR, USA, June. Association for Computational Linguistics.

Christine L. Borgman and Jonathan Furner. 2002. Scholarly communication and bibliometrics. *ANNUAL REVIEW OF INFORMATION SCIENCE AND TECHNOLOGY*, 36(1):2–72.

Razvan Bunescu and Raymond Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 724–731, Vancouver, British Columbia, Canada, October. Association for Computational Linguistics.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27.

J. Cohen. 1968. Weighted kappa: Nominal scale agreement with provision for scaled disagreement or partial credit. *Psychological Bulletin*, 70:213–220.

Aaron Elkiss, Siwei Shen, Anthony Fader, Güneş Erkan, David States, and Dragomir Radev. 2008. Blind men and elephants: What do citation summaries tell us about a research article? *J. Am. Soc. Inf. Sci. Technol.*, 59(1):51–62.

E. Garfield, Irving H. Sher, and R. J. Torpie. 1984. *The Use of Citation Data in Writing the History of Science*. Institute for Scientific Information Inc., Philadelphia, Pennsylvania, USA.

Claire Grover, Colin Matheson, Andrei Mikheev, and Marc Moens. 2000. Lt ttt - a flexible tokenisation tool. In *In Proceedings of Second International Conference on Language Resources and Evaluation*, pages 1147–1154.

Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. 2002. Gene selection for cancer classification using support vector machines. *Mach. Learn.*, 46:389–422, March.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18.

T. L. Hodges. 1972. Citation indexing-its theory and application in science, technology, and humanities. *Ph.D. thesis, University of California at Berkeley.Ph.D. thesis, University of California at Berkeley.*

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *IN PROCEEDINGS OF THE 41ST ANNUAL MEETING OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS*, pages 423–430.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

J. Richard Landis and Gary G. Koch. 1977. The Measurement of Observer Agreement for Categorical Data. *Biometrics*, 33(1):159–174, March.

Terttu Luukkonen. 1992. Is scientists' publishing behaviour rewardseeking? *Scientometrics*, 24:297–319. 10.1007/BF02017913.

Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

Qiaozhu Mei and ChengXiang Zhai. 2008. Generating impact-based summaries for scientific literature. In *Proceedings of ACL-08: HLT*, pages 816–824, Columbus, Ohio, June. Association for Computational Linguistics.

Saif Mohammad, Bonnie Dorr, Melissa Egan, Ahmed Hassan, Pradeep Muthukrishan, Vahed Qazvinian, Dragomir Radev, and David Zajic. 2009. Using citations to generate surveys of scientific paradigms. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 584–592, Boulder, Colorado, June. Association for Computational Linguistics.

Preslav I. Nakov, Ariel S. Schwartz, and Marti A. Hearst. 2004. Citances: Citation sentences for semantic analysis of bioscience text. In *In Proceedings of the SIGIR04 workshop on Search and Discovery in Bioinformatics*.

Hidetsugu Nanba and Manabu Okumura. 1999. Towards multi-paper summarization using reference information. In *IJCAI '99: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 926–931, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Hidetsugu Nanba, Noriko Kando, Manabu Okumura, and Of Information Science. 2000. Classification of research papers using citation links and citation types: Towards automatic review article generation.

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

Vahed Qazvinian and Dragomir R. Radev. 2010. Identifying non-explicit citing sentences for citation-based summarization. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 555–564, Uppsala, Sweden, July. Association for Computational Linguistics.

Vahed Qazvinian, Dragomir R. Radev, and Arzucan Ozgur. 2010. Citation summarization through keyphrase extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 895–903, Beijing, China, August. Coling 2010 Organizing Committee.

Dragomir R. Radev, Pradeep Muthukrishnan, and Vahed Qazvinian. 2009. The acl anthology network corpus. In *NLPIR4DL '09: Proceedings of the 2009 Workshop on Text and Citation Analysis for Scholarly Digital Libraries*, pages 54–61, Morristown, NJ, USA. Association for Computational Linguistics.

Ariel Schwartz, Anna Divoli, and Marti Hearst. 2007. Multiple alignment of citation sentences with conditional random fields and posterior decoding. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 847–857.

Advaith Siddharthan and Simone Teufel. 2007. Whose idea was this, and why does it matter? attributing scientific work to citations. In *In Proceedings of NAACL/HLT-07*.

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

Simone Teufel. 2007. Argumentative zoning for improved citation indexing. computing attitude and affect in text. In *Theory and Applications, pages 159170*.

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