Identifying Citing Sentences in Research Papers Using Supervised Learning

Kazunari Sugiyama
Department of Computer Science
National University of Singapore
Singapore
sugiyama@comp.nus.edu.sg

Tarun Kumar
Indian Institute of
Information Technology
Allahabad, India
tar.iiita@gmail.com

Min-Yen Kan
Department of Computer Science
National University of Singapore
Singapore
kanmy@comp.nus.edu.sg

Ramesh C. Tripathi
Indian Institute of
Information Technology
Allahabad, India
rctripathi@iiita.ac.in

***Abstract* — Researchers have largely focused on analyzing citation links from one scholarly work to another. Such *citing sentences* are an important part of the narrative in a research article. If we can automatically identify such sentences, we can devise an editor that helps suggest when a particular piece of text needs to be backed up with a citation or not. In this paper, we propose a method for identifying citing sentences by constructing a classifier using supervised learning. Our experiments show that simple language features such as proper nouns and the labels of previous and next sentences are effective features to identifying citing sentences.**

***Keywords* — information retrieval; digital library; discourse processing; citation analysis**

I. Introduction

When we write research papers or articles, we often make references to previous works in our own research field. Citations serve various purposes: as evidence for claims, as an acknowledgment of other's work, among other functions. We term sentences that contain such references as *citing sentences*. These statements are usually followed by a pointer to the full reference located at the end of a paper in a "Reference" or "Bibliography" section. Our aim in this work is to identify whether a sentence in a paper needs a citation or not. For example, consider the following two sentences:

Sentence 1:

"We want to build a system which can help in finding a person's job easily."

Sentence 2:

"The HSQL project was led by the Swedish State Bureau with participants from Sweden, Denmark, Finland, and Norway.

In the above two statements, Sentence 2 needs a citation because it refers to the previous work, namely, the HSQL project. On the other hand, most people would agree that Sentence 1 does not need a citation because there is no description of previous work.

In this paper, we propose a method[1] for identifying sentences that require citations such as Sentence 2 above. While citing sentences are often trivially marked with a citation marker (e.g., "[5]" or "(Brown, 1990)"), an important distinction in our problem is that we consider detecting such a sentence when such markers are not present. We show that our approach, which constructs a supervised classifier from simple features, achieves a high level of accuracy. With respect to references and other works on the analysis of citation information [1, 2], to the best of our knowledge, the work presented here is the first work to identify citing sentences using natural language analysis. Such a module is useful in a smart authoring environment, which can help authors by suggesting whether statements made in the paper draft need citations or not. Such a system will take in a research paper as input and identify the statements that need citation. It can also be used at the time of writing a paper by suggesting to authors whether current present statement needs a citation or not.

This paper is organized as follows: In Section II, we review related work on analyzing citations in research papers and briefly describe the two classifiers we use in our experiment. In Section III, we describe our approach in distinguishing between sentence that require citations and those that do not. In Section IV, we present the experimental results for evaluating our approach. Finally, we conclude the paper with a summary and directions for future work in Section V.

II. Related Work

We first review related works on scholarly citation analysis, and then survey the fundamental background of two state-of-the-art classifiers – maximum entropy and support vector machines – that we employ in later our experiments.

A. Citation Analysis of Research Papers

Citation information has been used for information retrieval since the early stages of this field. As far as we know, there are two types of research in the field of citation analysis of research papers, (1) citation count to evaluate the impact of scientific papers, and (2) citation context analysis.

Citation count is widely used in evaluating the importance of a paper because it is strongly correlated with academic document impact [3]. The Thomson Scientific ISI Impact Factor (ISI IF) is the representative approach using citation count [4], which factors citation count with a moving window to calculate the impact of certain publication venues. The advantages of citation count are (i) its simplicity in computation; and (ii) its proven track record in deployment in scientometric applications. However, citation count has well-known limitations: Citing papers with high impact and ones with low impact are treated equally in standard citation count.

[1] This work is supported by a Media Development Authority (MDA) grant "Interactive Media Search," R-252-000-325-279.

In order to overcome this problem, many works recently have employed the notion of PageRank [5] to better weight and control for the influence of papers of differing impact [6, 7, 8, 9, 10].

Citation information is statistical in nature. Therefore, many researchers have focused on this characteristic. For example, Kessler et al. [11] proposed to use the notion of *bibliographic coupling*, where two documents are said to be coupled if they share one or more references. Small [12] proposed a complementary method, termed co-citation analysis, where the similarity between documents A and B is measured by the number of documents that cite A and B. In addition, researchers have also focused on the potential usefulness of the text associated with citations in specific applications, such as text summarization [13, 14], thesaurus construction [15], and information retrieval [16, 17].

B. Maximum Entropy (ME)

The framework of maximum entropy [18] has already been widely used for a variety of natural language tasks such as prepositional phrase attachment [19], language modeling [20, 21], part-of-speech tagging [22] and text segmentation [23]. Maximum entropy has been shown to be an effective and competitive algorithm in these domains.

Statistical modeling constructs a model that best accounts for some training data. Specifically, for a given empirical probability distribution $\tilde{p}$, a model p is built to result in a distribution as close to $\tilde{p}$ as possible. Given a set of training data, there are numerous ways to choose a model p that accounts for the data. It can be shown that the probability distribution defined by Equation (1) is the one that is closest to $\tilde{p}$ in the sense of Kullback-Leibler divergence, when subjected to a set of feature constraints:

$$P(y \mid x) = \frac{1}{Z(x)} \exp\left[\sum_{i=1}^{k} \lambda_i f_i(x,y)\right], \qquad (1)$$

where $p(y \mid x)$ denotes the conditional probability of predicting an *class y* on seeing the *context x*. $f_i(x,y)$ $(i = 1,\cdots,k)$ are feature functions, λ_i $(i = 1,\cdots,k)$ are the weighting parameters for $f_i(x,y)$ $(i = 1,\cdots,k)$. k is the number of features and $Z(x)$ is a normalization factor to ensure that the $p(y \mid x)$ scores sum to one and reflect true probabilities. This maximum entropy model represents evidence with binary functions known as *contextual predicates* in the form:

$$f_{cp,y'}(x,y) = \begin{cases} 1 & \text{If } y=y' \text{ and } cp(x)=true \\ 0 & \text{otherwise} \end{cases}$$

where *cp* is the *contextual predicate* that maps a *outcome y* and *context x* pair to {*true, false*}.

The human expert can choose arbitrary feature functions in order to reflect the characteristics of the problem domain as faithfully as possible. The ability of freely incorporating problem-specific knowledge in terms of feature functions gives ME models an advantage over other learning paradigms, which often suffer from strong feature independence assumptions (e.g., in the case of the naïve Bayes classifier).

Once a set of features is chosen by the human expert, the corresponding maximum entropy model can be constructed by adding features as constraints to the model and iteratively adjusting the weights of these features automatically to best reflect the training data. Formally, we require that:

$$E_{\tilde{p}} < f_i >= E_p < f_i >,$$

where $E_{\tilde{p}} < f_i >= \sum_x \tilde{p}(x,y) f_i(x,y)$ is the empirical expectation with respect to the model distribution *p*. Among all the models subjected to these constraints, a unique solution exists that preserves the uncertainty in the original constraints and does not add any extra bias to the solution – this is the maximum entropy solution obtained by the training procedure.

Given an exponential model with *n* features and a set of training data (empirical distribution), we need to find the associated weight for each of the *n* features to maximize the model's log-likelihood:

$$L(p) = \sum_{x,y} \tilde{p}(x,y) \log p(y \mid x).$$

It is important to select an optimal model subjected to given constraints from the log-linear family. There are three well-known iterative scaling algorithms specially designed to estimate parameters of ME models of the Equation (1): Generalized Iterative Scaling [24] and Improved Iterative Scaling [25], and Limited–Memory Variable Metric [26].

C. Support Vector Machine (SVM)

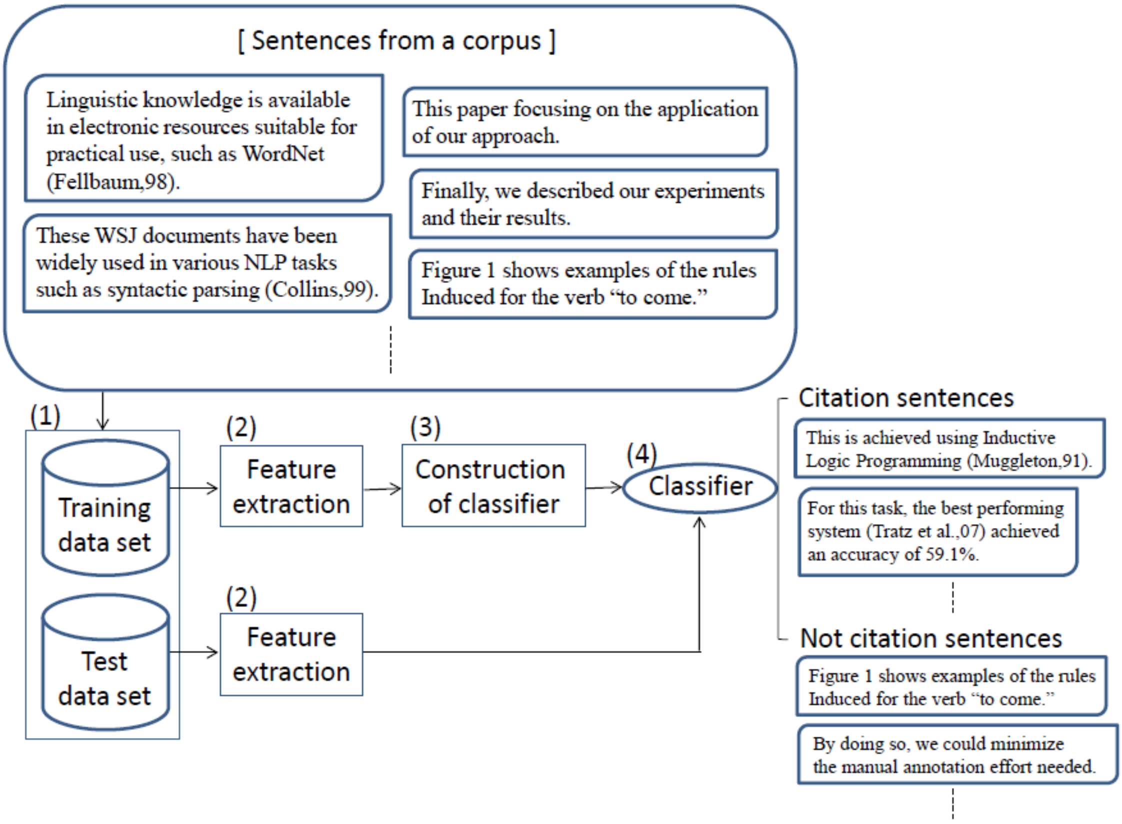
Support Vector Machine (SVM) [27] has many desirable qualities that make it one of the most popular algorithms. It not only has a solid theoretical foundation, but also classifies more accurately than most other algorithms in many applications such as Web page classification and bioinformatics tasks.

Given a training set of instance label pairs ($\mathbf{x}_i$, y_i) (i=1,…, l) where $\mathbf{x}_i \in R^n$ is a training vector and $y_i \in \{1,-1\}^l$ is its class label, an SVM finds a linear separating hyperplane with the maximal margin as a solution to the following optimization problem:

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i \qquad (2)$$

$$\text{subject to } y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \geq 1 - \xi_i,\ \xi_i \geq 0.$$

As the original problem may not be linearly separable, $\mathbf{x}_i$ can be mapped into a higher dimensional space by a function ϕ. Then, SVM finds a linear separating hyperplane with the maximal margin in this higher dimensional space. $C > 0$ is the penalty parameter of the error term. Interestingly, $K(\mathbf{x}_i,\mathbf{x}_j) \equiv \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$, the kernel function, can be of differing forms – linear, polynomial, radial basis and, sigmoid functions are often used. The SVM depends directly on the kernel function, and if a surrogate method that yields the function values can be given, the explicit Cartesian product need not be calculated, greatly saving computational complexity.



Linguistic knowledge is available in electronic resources suitable for practical use, such as WordNet (Fellbaum,98).

These WSJ documents have been widely used in various NLP tasks such as syntactic parsing (Collins,99).

This paper focusing on the application of our approach.

Finally, we described our experiments and their results.

Figure 1 shows examples of the rules Induced for the verb "to come."

This is achieved using Inductive Logic Programming (Muggleton,91).

For this task, the best performing system (Tratz et al.,07) achieved an accuracy of 59.1%.

Figure 1 shows examples of the rules Induced for the verb "to come."

By doing so, we could minimize the manual annotation effort needed.

Figure 1. Overview of our system.

III. Proposed Method

Figure 1 illustrates our proposed system. Our proposed system consists of the following four parts. We detail each of these steps individually.

(1) Constructing proper training and test data sets,
(2) Extracting the appropriate features from the data,
(3) Constructing the classifier,
(4) Classifying sentences as citing or non-citing.

(1) Constructing proper training and test data sets

For our experiments, we utilize the Association for Computational Linguistics' standard research article corpus, the ACL Anthology Reference Corpus (ACL ARC), discussed in Section IV. We first remove Reference section and stop words[2] from each paper. We define a sentence that contains citing information as positive instance, and a sentence that does not contain that as negative instance. Our idea is to use sentences that have existing citations as training data, by removing the citation marker. If a citation marker is found via heuristic rules, we remove it. We then divide our dataset into ten equal sized parts, to be used as cross validation folds. As we perform 10-fold cross validation, we divided the whole data set into 90% of training data and 10% of test data and repeat the below experimental process 10 times to obtain our final evaluation results.

(2) Extracting the appropriate features from the data

In this module, we extract features from each sentence in order to construct the classifier later. The features we extracted are as follows:

a) Unigram - Unigram features include the words contained in the sentences. After removing the stop words from the sentences, each word appearing in the sentence serves as a binary feature, turned on for the individual sentence. Unigrams are an important class of features as certain types of words tend to appear more frequently in citing/non-citing sentence. For example, the sentence, "The classification model is trained …" can be represented with the unigrams "classification," "model," "trained" ("the" and "is" are stop words), among others.

b) Bigram - By combining two adjacent words in sentences, we create bigram features. Bigram features help in analyzing the effect of two independent words in combination. This combination helps significantly in classification. Using the same example as above, we extract the bigrams, "classification model," "model trained" (similarly, "the" and "is" are stop words), among others.

c) Proper Nouns - These are nouns that give the names of people, locations, systems and organizations. Based on the presence of various types of proper nouns, the respective binary features are set. This feature also

[2] List of 571 words obtained from ftp://ftp.cs.cornell.edu/pub/smart/english.stop

plays a significant role in detecting citing sentences, as from experience, we know that such sentences often refer to particular scholars, their developed systems or institutions.

d) Previous and Next Sentence - We also can include information about the classifications of neighboring sentences – their citation/non-citation status. For example, if the previous sentence is a citing sentence, the following sentence may continue to discuss the same work and would be less likely to contain an additional citation.

e) Position - The positional feature gives information about the part of document in which the sentence appears. To implement this feature, we divide the document into six equal parts: one part for the first 1/6th of the document, one part for the second 1/6th, until the final sixth 1/6th. We turn on one of the six binary features based on which part of document the sentence appears in. These features are important as statements appearing in certain sections exhibit markedly different probabilities for citation. For example, sentences in the middle or end of a research paper are likely to discuss the authors' own work or the evaluation and are less probable citation areas, as compaerd to the beginning of the article, where authors often discuss and credit prior work.

f) Orthographic - This set of features check for miscellaneous formatting characteristics, including specific orthographies used in the sentence. For example, a sentence containing numbers or single capital letters may be more indicative of citing sentences, as they may present comparative results or author initials from cited works.

(3) Constructing the classifier

Using the training data set described in (1), we construct two supervised classifiers based on two publicly available implementations of supervised learning frameworks: maximum entropy (ME)[3] [18] and support vector machine (SVM)[4] [27] as described in Section II. Both methodologies were chosen as they efficiently handle a large number of non-independent features, common to many natural language tasks. In addition, our task is a binary classification problem – whether a sentence is citing sentence or not. SVM often brings superior results in binary classification tasks.

(4) Classifying sentences as citing or non-citing

Given trained classifiers in (3), we can then apply the trained models to new, unseen sentences. We employ these trained models and assess the performance of the models using "accuracy" as our evaluation measure.

IV. Experiments

A. *Experimental Data*

We used the ACL Anthology Reference Corpus (ACL ARC)[5] [28]. The ACL ARC is constructed from a significant subset of the ACL Anthology[6], which is a digital archive of conference and journal papers in natural language processing and computational linguistics. The ACL ARC consists of 10,921 articles from the February 2007 snapshot of the ACL Anthology. Using the ACL ARC, we extracted features from each of the 955,755 sentences included in the 10,921 articles. Using regular expression pattern matching to find citation markers, we identified 112,533 sentences as citing sentences (positive instances), and post-processed them to remove the citation marker. We deemed the remaining 843,242 sentences as non-citing sentences (negative instances). In our experiments, the aim is to identify whether each sentence is a citing sentence or not.

B. *Experimental Results*

Using each of the individual feature classes from *a)* to *f)* described in Section III, we constructed both SVM and ME classifiers. We evaluated our models using simple accuracy defined as follows:

$$Accuracy = \frac{\text{(Number of correct classifications)}}{\text{(Total number of test cases)}},$$

where correct classifications means that the learned model predicts the same class as the original class of the test case.

1) Classification Accuracy by Cross Validation

We first conducted experiments using 10-fold cross validation for both ME and SVM. For the SVM experiments, we used the default settings of LIBSVM package, setting the kernel as the radial basis function, and the value of C in Equation (2) as 1.0. Table 1 shows experimental results obtained using classifiers constructed both ME and SVM. For comparison, we also constructed an integrated classifier which uses all of six features, labeled as "All" in Table 1.

Table 1. Accuracy obtained by ME and SVM.

Feature	Accuracy (ME)	Accuracy (SVM)
(1) Unigram	0.876	0.879
(2) Bigram	0.827	0.851
(3) Proper Noun	**0.882**	**0.882**
(4) Previous and Next Sentence	**0.882**	**0.882**
(5) Position	0.875	0.877
(6) Orthographic	0.878	0.880
(7) All [(1) - (6)]	0.876	0.878

[3] Maximum Entropy Modeling Toolkit (Version 20041229), http://homepages.inf.ed.ac.uk/lzhang10/maxent_toolkit.html

[4] LIBSVM (Version 2.89), http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[5] Version 20080325, http://acl-arc.comp.nus.edu.sg/

[6] http://www.aclweb.org/anthology/

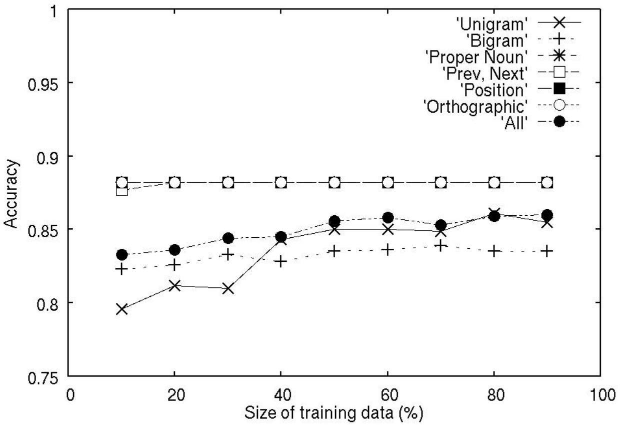


Figure 2. Classification accuracy obtained by ME.

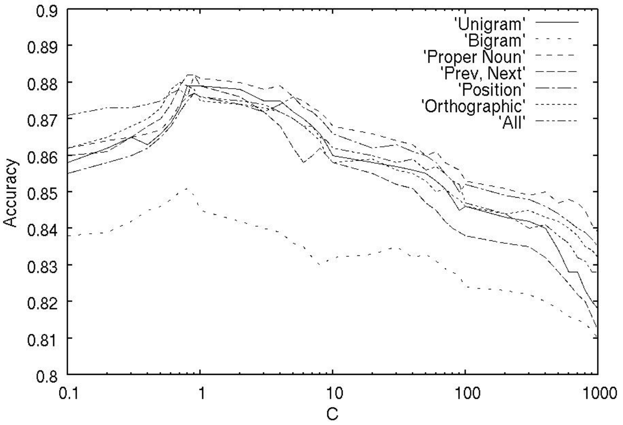


Figure 4. Classification accuracy obtained by SVM with different value of *C*.

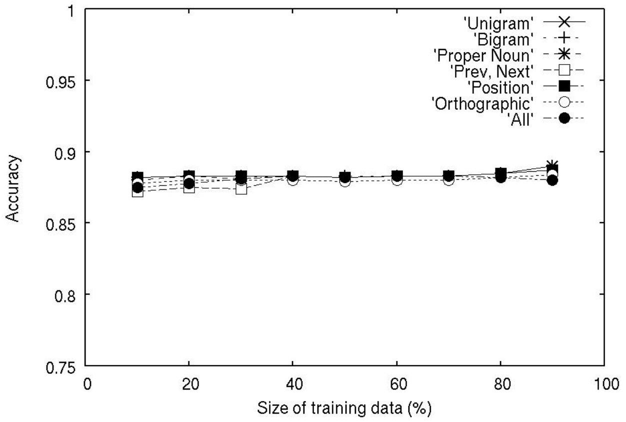


Figure 3. Classification accuracy obtained by SVM.

Table 2. Optimal values of *C* and their accuracy.

Feature	Optimal Value of *C*	Accuracy
(1) Unigram	0.8	0.879
(2) Bigram	0.8	0.851
(3) Proper Noun	0.9	**0.882**
(4) Previous and Next Sentence	0.9	**0.882**
(5) Position	0.9	0.877
(6) Orthographic	0.9	0.880
(7) All [(1) - (6)]	0.9	0.878

2) Classification Accuracy on Different Size of Training Data

For most of learning algorithms, the size of the training data affects the classification accuracy. Therefore, we conducted experiments to empirically assess how classification performance changes when the size of training data is a subset of the full training data – from 10% to 90%. Figures 2 and 3 (both shown at the same scale) give the experimental results obtained by ME and SVM, respectively.

3) Classification Accuracy in Different Value of "C" in SVM

Finally, we conducted a series of experiments in tuning the SVM performance. In the SVM framework, the value of *C* in Equation (2) — the error term — denotes the tolerance of the SVM to accept misclassifications in the separating hyperplane, also affects classification accuracy. Thus, we conducted experiments to find the classification accuracy obtained by different values of *C*. Figure 4 shows the accuracy obtained by using SVM with different values of *C*.

C. Discussion

According to Table 1, in the first set of experiments on cross validation, both ME and SVM achieved an accuracy greater than 0.82. We observed a small difference of accuracy (0.002 to 0.024) between these classifiers. Therefore, we can find that the accuracy of this kind of task does not depend on classifiers. Especially, simple features such as "Proper Noun" and the context of "Previous and Next Sentence" bring better results (0.882) among them. Interestingly, the "Bigram" feature is not so effective among the features we used. As bigram features are often very sparse, it may be difficult to construct an accurate classifier with such few overlapping features.

By varying the size of the training data, we obtain slight variations in performance in both the SVM and ME frameworks. According to Figure 2, ME exhibits more performance variation, where the accuracy of features such as "Unigram," "Bigram," and "All" is influenced by the size of training data. In other words, the larger the size of the training data, the more accurate the classification results are.

According to Figure 3, the SVM framework shows less variation. When the data size was slightly reduced (80-90% of the original), a slight improvement in accuracy is observed. For example, in "Previous and Next Sentence," while the accuracy at 10% of training data is 0.872, the accuracy at 90% of training data is 0.882. Moreover, in "All," while the accuracy at 10% of training data is 0.875, the accuracy at 90% of training data is 0.880. In future work, we may investigate this peculiarity in more detail.

Finally, according to Figure 4, when different error term values of C in the SVM framework were used, we observed that the best accuracy is obtained when the value of C is set to slightly lower than 1.0 in each feature. The optimal values of C and their accuracy obtained by each feature are shown in Table 2. Together with the first experimental results, the results show that "Proper Noun" and the context of "Previous and Next Sentence" features bring the best accuracy in both ME and SVM (0.882 with C=0.9). Surprisingly, the composite classifiers that use all feature classes underperform classifiers that are trained only on these two sources of data.

V. Conclusion

We have described a method for identifying citation sentences by constructing classifier using supervised learning approaches with simple features extracted from research papers. Experimental results showed that both proper nouns and contextual classification of the previous and next sentence are effective features for training accurate models in both SVM and ME frameworks. In future work, we plan to build an editor that will help authors write a research paper by advising them when statements in their draft need a citation or not.

References

[1] S. Lawrence, C. L. Giles, and K. Bollacker: "Digital Libraries and Autonomous Citation Indexing," *IEEE Computer*, 32(6): 67-71, 1999.

[2] I. G. Councill, C. L. Giles, and M.-Y. Kan: "ParsCit: An Open-Source CRF Reference String Parsing Package," In *Proc. of* the *6th International Conference on Language Resources and Evaluation Conference (LREC08),* pages 661-667, 2008.

[3] F. Narin: "Evaluative Bibliometrics: The Use of Publication and Citation Analysis in the Evaluation of Scientific Activity," *Computer Horizons*, Cherry Hill, NJ, 1976.

[4] E. Garfield: "Citation Indexing: Its Theory and Application in Science, Technology, and Humanities," John Wiley and Sons, NY, 1979.

[5] L. Page, S. Brin, R. Motwani, and T. Winograd: "The PageRank Citation Ranking: Bringing Order to the Web," Stanford Digital Library Technologies Project, *SIDL-WP-1999-0120*, 1998.

[6] J. Bollen, M. A. Rodriguez and H. Van De Sompel: "Journal Status," *Scientometrics*, 69(3): 669-687, 2006.

[7] Y. Sun and C.L. Giles: "Popularity Weighted Ranking for Academic Digital Libraries," In *Proc. of the 29th European Conference on Information Retrieval (ECIR 2007)*, pages 605-612, 2007.

[8] M. Krapivin and M. Marchese: "Focused PageRank in Scientific Papers Ranking," In *Proc. of the 11th International Conference on Asian Digital Libraries (ICADL 2008)*, Lecture Notes in Computer Science (LNCS), Vol. 5362, pages 144-153, 2008.

[9] N. Ma, J. Guan, and Y. Zhao: "Bringing PageRank to the Citation Analysis," *Information Processing and Management*, 44(2), pages 800-810, 2008.

[10] H. Sayyadi and L. Getoor: "FutureRank: Ranking Scientific Articles by Predicting their Future PageRank," In *Proc. of the 9th SIAM International Conference on Data Mining*, pages 533-544, 2009.

[11] M. M. Kessler: "Bibliographic Coupling Between Scientific Papers," *American Documentation*, 14(1): 10-25, 1963.

[12] H. Small: "Co-Citation in the Scientific Literature: A New Measure of the Relationship Between Two Documents," *Journal of the American Society of Information Science*, 24(4):265-269, 1973.

[13] S. Teufel, A. Siddharthan, and D. Tidhar: "Automatic Classification of Citation Function," In *Proc. of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, pages 103-110, 2006.

[14] V. Qazvinian and D.R. Radev: "Scientific Paper Summarization Using Citation Summary Networks," In *Proc. of the 22nd International Conference on Computational Linguistics (Coling2008)*, pages 689-696, 2008.

[15] J. Schneider: "Verification of Bibliometric Methods' Applicability for Thesaurus Construction," PhD thesis, Royal School of Library and Information Studies, 1994.

[16] A. Ritchie, S. Teufel, and S. Robertson: "Using Terms from Citations for IR: Some First Results," In *Proc. of the 29th European Conference on Information Retrieval (ECIR 2007)*, pages 211- 221, 2007.

[17] A. Ritchie, S. Robertson and S. Teufel: "Comparing Citation Contexts for Information Retrieval," In *Proc. of the 17th International Conference on Information and Knowledge Management (CIKM'08)*, pages 213-222, 2008.

[18] A. L. Berger, S. A. Della Pietra, and V. J. Della Pietra: "A Maximum Entropy Approach to Natural Language Processing," *Computational Linguistics*, 22(1):39-71, 1996.

[19] A. Ratnaparkhi, J. Reynar, and S. Roukos: "A Maximum Entropy Model for Prepositional Phrase Attachment," In *Proc. of the ARPA Human Language Technology Workshop*, pages 250-255, 1994.

[20] R. Rosenfeld: "Adaptive Statistical Language Modeling: A Maximum Entropy Apporach," PhD thesis, Carnegie Mellon University, 1994.

[21] S. F. Chen and R. Rosenfeld: "A Gaussian Prior for Smoothing Maximum Entropy Models," Technical Report *CMU-CS-99-108*, Carnegie Mellon University, 1999.

[22] A. Ratnaparkhi: "A Maximum Entropy Model for Part-Of-Speech Tagging," In *Proc. of the Conference on Empirical Methods in Natural Language Processing*, pages 133-142, 1996.

[23] D. Beeferman, A. Berger, and J. Lafferty: "Statistical Models For Text Segmentation," *Machine Learning*, 34(1-3):177-210, 1999.

[24] J. N. Darroch and D. Ratcliff: "Generalized Iterative Scaling for Log-Linear Models," *The Annals of Mathematical Statistics*, 43(5):1470-1480, 1972.

[25] S. D. Pietra, V. J. Della Pietra, and J. D. Lafferty: "Inducing Features of Random Fields." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4): 380-393, 1997.

[26] R. Malouf: "A Comparison of Algorithms for Maximum Entropy Parameter Estimation," In *Proc. of the 6th Conference on Natural Language Learning (CoNLL-2002)*. pages 49–55, 2002.

[27] V. Vapnik: "The Nature of Statistical Learning Theory," Springer, NY, 1995.

[28] S. Bird, R. Dale, B. J. Dorr, B. Gibson, M. T. Joseph, M.-Y. Kan, D. Lee, B. Powley, D. R. Radev, Y. F. Tan: "The ACL Anthology Reference Corpus: A Reference Dataset for Bibliographic Research in Computational Linguistics," In *Proc. of the 6th International Conference on Language Resources and Evaluation Conference (LREC08),* pages 1755-1759, 2008.