BMC Bioinformatics

RESEARCH Open Access

A structural SVM approach for reference parsing

Xiaoli Zhang*, Jie Zou, Daniel X Le, George R Thoma

From Machine Learning for Biomedical Literature Analysis and Text Retrieval in the International Conference for Machine Learning and Applications 2010
Washington, DC, USA. 12-14 December 2010

Abstract

Background: Automated extraction of bibliographic data, such as article titles, author names, abstracts, and references is essential to the affordable creation of large citation databases. References, typically appearing at the end of journal articles, can also provide valuable information for extracting other bibliographic data. Therefore, parsing individual reference to extract author, title, journal, year, etc. is sometimes a necessary preprocessing step in building citation-indexing systems. The regular structure in references enables us to consider reference parsing a sequence learning problem and to study structural Support Vector Machine (structural SVM), a newly developed structured learning algorithm on parsing references.

Results: In this study, we implemented structural SVM and used two types of contextual features to compare structural SVM with conventional SVM. Both methods achieve above 98% token classification accuracy and above 95% overall chunk-level accuracy for reference parsing. We also compared SVM and structural SVM to Conditional Random Field (CRF). The experimental results show that structural SVM and CRF achieve similar accuracies at token- and chunk-levels.

Conclusions: When only basic observation features are used for each token, structural SVM achieves higher performance compared to SVM since it utilizes the contextual label features. However, when the contextual observation features from neighboring tokens are combined, SVM performance improves greatly, and is close to that of structural SVM after adding the second order contextual observation features. The comparison of these two methods with CRF using the same set of binary features show that both structural SVM and CRF perform better than SVM, indicating their stronger sequence learning ability in reference parsing.

Background

Bibliographic references, typically cited at the end of scientific articles, provide much valuable information. Parsing these references is an essential step for building citation-indexing systems. Many well-known citation-indexing systems, such as CiteSeer [1], ISI Web of Knowledge [2] and Google Scholar [3], could have implemented complex reference parsing algorithms, though detailed reports about their algorithms and performance have not been found in the literature. As the authors of CiteSeer mention in [4], the reliable parsing of references may still be considered an open problem.

MEDLINE®, the flagship database of the U.S. National Library of Medicine, contains over 18 million citations to the medical journal literature and is a critical source of information for biomedical research and clinical medicine. With the rapid increase of journal literature indexed by MEDLINE every year, it is essential to have automated methods to extract bibliographic data, including article titles, author names, affiliations, abstracts, and many others.

While references are not included in MEDLINE citations, they are indispensable for detecting several other items. For example, creating the Comment-On/Comment-In field for MEDLINE (identifying pairs of articles, with one article commenting on the other) requires matching references to the citing text [5]. In addition, assigning Medical Subject Heading (MeSH) terms [6],

* Correspondence: zhangxiaol@mail.nih.gov
Lister Hill National Center for Biomedical Communications, National Library of Medicine, 8600 Rockville Pike, Bethesda, MD 20894, USA
Full list of author information is available at the end of the article

an essential step in indexing the article, may also benefit from analyzing the MeSH terms assigned to the cited articles, which requires parsing the references to those articles. Reliable reference parsing is therefore an important step for automatically creating citations for MEDLINE.

In this work, our goal is to extract the following 7 entities from the references: Citation Number (<N>), Author Names (<A>), Article Title (<T>), Journal Title (<J>), Volume (<V>), Pagination (<P>) and Publication Year (<Y>). All remaining words in the reference are labeled as Other (<O>). The notation inside each parenthesis is the abbreviated entity label.

In the large number of journals (over 5,200) indexed for MEDLINE, references are formatted in a large variety of ways, some of which are shown in Table 1. In each example, the original reference is followed by the ground-truth labeling. Most of the references cite "normal" journal articles, but a small number cite books, e.g., (f) and international standards, e.g., (g). Some references omit Citation Numbers, e.g., (c), and among others which do have these, there are different formats either as a single number or an author-year chunk, e.g., (a) and (b). There is also some variation in the way Author Names are expressed: initials followed by last names, e.g. (a); last name followed by initials, e.g., (d); not all authors listed, e.g., (e); the first author and the remaining authors in different formats, e.g., (c); and occasionally an anonymous author, e.g., (g). Most Journal Titles are significantly abbreviated, and most Paginations consist only of digits, but (d) is an example where Pagination contains non-digit characters. There are also many variations in the use of commas, spaces, semicolons or periods to separate different entities; and in character capitalizations. This wide variability makes reliable reference parsing a challenging task.

Early research in reference parsing involved rule-based methods, which usually depend on knowledge that is manually crafted and based on a domain expert's observation. This domain knowledge is organized as templates or hierarchical frameworks, which summarize the recognizable patterns formed by the data or the surrounding text, and the rules associated with those recognizable patterns. After the knowledge representation is built, various algorithms can be used to match the text to the knowledge representation, and to extract data according to the rules. These matching algorithms include template mining [7,8], INFOMAP [9,10] and BLAST (Basic Local Alignment Search Tool), a tool originally designed for gene sequence alignment [11].

Rule-based methods can be very successful when the references are from a small or moderate number of journals. This is because journal publishers usually require authors to strictly follow predefined citation

Table 1 Examples of references following different styles in medical journal articles

(a) 19 S. Miyazaki, K. Takahashi, M. Shiraki, T. Saito, Y. Tezuka and K. Kasuya, Properties of a poly(3-hydroxybbutyrate) depolymerase from Penicillium funiculosum, J. Polym. Environ. 8 (2002), pp. 175–182.

<N>19</N> <A>S. Miyazaki, K. Takahashi, M. Shiraki, T. Saito, Y. Tezuka, K. Kasuya,</A> <T>Properties of a poly(3-hydroxybbutyrate) depolymerase from Penicillium funiculosum,</T> <J>J. Polym. Environ.</J> <V>8</V> <Y>(2002),</Y> <P>pp. 175–182.</P>

(b) Sofuoglu and Kosten, 2005 M. Sofuoglu and T.R. Kosten, Novel approaches to the treatment of cocaine addiction, CNS Drugs 19 (2005), pp. 13–25. Full Text via CrossRef | Abstract + References in Scopus | Cited By in Scopus

<N>Sofuoglu and Kosten, 2005</N> <A>M. Sofuoglu and T.R. Kosten,</A> <T>Novel approaches to the treatment of cocaine addiction,</T> <J>CNS Drugs</J> <V>19</V> <Y>(2005),</Y> <P>pp. 13–25.</P> <O>Full Text via CrossRef | Abstract + References in Scopus | Cited By in Scopus</O>

(c) Czarnetzki, A. B., and C. C. Tebbe. 2004. Diversity of bacteria associated with Collembola: a cultivation-independent survey based on PCR-amplified 16S rRNA genes. FEMS Microbiol. Ecol. 49:217-227.[CrossRef]

<A>Czarnetzki, A. B., and C. C. Tebbe.</A> <Y>2004.</Y> <T>Diversity of bacteria associated with Collembola: a cultivation-independent survey based on PCR-amplified 16S rRNA genes.</T> <J>FEMS Microbiol. Ecol.</J> <V>49:</V> <P>217-227.</P> <O>[CrossRef]</O>

(d) Rios R, Carneiro I, Arce VM, and Devesa J. Myostatin is an inhibitor of myogenic differentiation. Am J Physiol Cell Physiol 282: C993–C999, 2002. [Abstract/Free Full Text]

<A>Rios R, Carneiro I, Arce VM, and Devesa J.</A> <T>Myostatin is an inhibitor of myogenic differentiation.</T> <J>Am J Physiol Cell Physiol</J> <V>282:</V> <P>C993–C999,</P> <Y>2002.</Y> <O>[Abstract/Free Full Text]</O>

(e) 12. T.J. McCarthy et al., Chem. Biol. 12, 1221 (2005). [CrossRef] [ISI] [Medline]

<N>12.</N> <A>T.J. McCarthy et al.,</A> <J>Chem. Biol.</J> <V>12,</V> <P>1221</P > <Y>(2005).</Y> <O>[CrossRef] [ISI] [Medline]</O>

(f) 18 J. Cavanagh, W.J. Fairbrother, A.G. Palmer and N.J. Skelton, Protein NMR Spectroscopy, Academic Press, San Diego, CA (1996).

<N>18</N> <A>J. Cavanagh, W.J. Fairbrother, A.G. Palmer and N.J. Skelton,</A> <J>Protein NMR Spectroscopy,</J> <O>Academic Press, San Diego, CA</O> <Y>(1996)</Y>

(g) Anonymous. 2005. Microbiology of food and animal feeding stuffs. Polymerase chain reaction (PCR) for the detection of food-borne pathogens. Requirements for amplification and detection for qualitative methods. Draft International Standard ISO/FDIS 20838:2005. DIN, Berlin, Germany.

<A>Anonymous.</A> <Y>2005.</Y> <T>Microbiology of food animal feeding stuffs. Polymerase chain reaction (PCR) for the detection of food-borne pathogens. Requirements for amplification detection for qualitative methods.</T> <O>Draft International Standard ISO/FDIS 20838</O> <Y>2005.</Y> <O>DIN, Berlin, Germany.</O>

styles, conduct careful editorial checking and correction before publishing. However, when a large number of journals are involved, it can be very challenging to build a sound knowledge representation due to the large variety of, and sometimes conflicting, citation styles. Rule-based methods also require domain experts to design the rules and maintain them over time, and therefore lack adaptability and are difficult to tune.

Machine learning approaches have recently attracted increased attention because they automatically learn the knowledge from training samples and therefore exhibit good adaptability. For example, Parmentier and Belaïd have developed a concept network to hierarchically represent and recognize structured data from bibliographic citations [12]. Besagni et al. took a bottom-up approach based on Part-of-Speech (PoS) tagging [13]. Basic tags, which are easily recognized, are first grouped into homogeneous classes. Confusing tokens are then classified by either a set of PoS correction rules or a structure model generated from well-detected tokens.

Reference parsing is essentially a sequence processing task and therefore statistical sequence models, e.g., *Hidden Markov Model* (*HMM*) and *Conditional Random Field* (*CRF*), as successful machine learning tools for information retrieval, have also been studied for parsing references. For example, Takasu applied HMM for metadata extraction from erroneous references [14]. Another frequently adopted machine learning method for information extraction is the *Support Vector Machine* (*SVM*) classifier. Okada et al. combined SVM and HMM for bibliographic component extraction [15]. In our previous research, we developed and compared a SVM-based method with one based on CRF [16].

Since collecting ground-truth training samples can be labor-intensive, unsupervised approaches have also been proposed. For example, Cortze et al. proposed an unsupervised approach, called FLUX-CiM, which is based on a frequency-tuned lexicon and includes four stages: blocking, matching, binding and joining [17].

There are also a few reference parsing libraries available online. These include ParsCit [18] and FreeCite [19].

As pointed out in a recent article, despite over a decade of research, reference parsing is still an unsolved task for several reasons, including data-entry errors, the wide variability of citation formats, lack of (or enforcement of) standards, large-scale citation data, and so on [4].

In this paper, we describe an extension of our previous work on reference parsing, reported in [16]. We adopted the recently proposed structural SVM method and compared it to conventional SVM. Our experiments on 1800 ground-truth labeled references show that the structural SVM method achieves over 98% token-level accuracy and over 95% chunk-level accuracy. In addition, we compared SVM and structural SVM to Conditional Random Field (CRF), another state-of-the-art sequence learning method. We observe that structural SVM and CRF achieve about the same accuracies at token- and chunk-levels. Both methods show the advantage of stronger sequence learning ability over SVM.

Methods

Mathematical description of structural SVM

Structural Support Vector Machine (*Structural SVM*), introduced by Tsochantaridis et al., is a supervised learning method designed for predicting complex structured outputs, such as sequences, trees and graphs [20]. Given a training sample of input-output pairs (x_1,y_1),...$(x_n,y_n)\in X \times Y$ drawn from an unknown distribution, structural SVM addresses the general problem of learning a mapping $f: X \to Y$ from input patterns $x \in X$ *to* discrete outputs $y \in Y$ that has low prediction errors. The idea is to learn a discriminant function F from which we can derive a prediction by maximizing F over Y given a specific input $x : f(x;w) = \arg\max_{y\in Y} F(x,y,w)$. $F(x,y,w) = w^T\Psi(x,y)$ is a linear combination of some joint feature representations of inputs and outputs, where w is a parameter vector and ψ is a feature vector relating x *and* y. The flexibility in designing ψ allows structural SVM to model many problems as diverse as natural language parsing, multiclass classification, sequence learning, etc.

Training the parameter vector w in structural SVM generalizes the maximum-margin principle in traditional SVM, leading to a quadratic optimization problem similar to multi-class SVM [20,21].

$$\begin{aligned}
\min_{w,\xi} \quad & \frac{1}{2}\|w\|^2 + \frac{C}{n}\sum_{i=1}^{n}\xi_i \\
s.t. \quad & w^T(\Psi(x_i,y_i) - \Psi(x_i,y)) \geq 1 - \xi_i \quad \forall y \in Y \setminus y_i,\ \forall i = 1,\cdots,n \\
& \xi_i \geq 0,\ \ i = 1,\cdots,n
\end{aligned}$$

The constraints are built upon the condition that given a training sample (x_i,y_i), the value of $w^T\Psi(x_i,y_i)$ for the correct prediction y_i should be greater than those for all other incorrect predictions y. Each training sample is associated with $|y|$ -1 constraints which share the same slack variable ξ_i. The introduction of ξ_i allows structural SVM to learn a large soft margin with small misclassification errors, which makes structural SVM more general to solve those classification problems where different classes are not strictly separable even in high feature space. The objective function is penalized by adding non-zero slack variables, ξ_i , each of which measures the degree of misclassification of a sample x_i. Therefore, the optimization becomes a trade-off between a large margin and a small error penalty. $\Sigma\xi_i$ gives an upper bound for the empirical risk on the training set, and the constant C is a regularization term that controls the trade-off between training error minimization and margin maximization. Training structural SVM is computationally expensive due to the large number of margin constraints. By an equivalent 1-slack reformulation of the n-slack structural SVM, Joachims et al. proposed a "l-slack cutting-plane" method which significantly reduces the computation time, thereby making the training on large databases feasible [21]. Both SVM and structural SVM are discriminative models. They learn optimal linear-separable hyperplanes with maximum-margin between classes. Structural SVM conducts global optimization on the whole structure, while SVM optimizes locally on individual tokens. Structural SVM is more general than SVM in its capability of learning interdependent and structured outputs. It has shown promising results for building highly complex, but still accurate discriminative models in the areas of classification with taxonomies, protein sequence alignment, and natural language context-free grammar parsing.

Feature extraction

A reference is first preprocessed and segmented into individual word tokens based on spaces and punctuations such as commas, periods, semi-colons, brackets, etc. We then extract 14 binary features and one normalized position feature from each token. They are briefly explained in Table 2. The first three are dictionary features which are collected by looking up a candidate word in Author Name, Article Title, and Journal Title dictionaries. We built these dictionaries from 10 years of MEDLINE data that contains about 236,748 Author Name words, 108,484 Article Title words, and 6,909 Journal Title words. The remaining 12 features provide further important information to help identify different entities.

Features from neighboring tokens are very informative as they exploit the contextual dependencies between tokens. There are two kinds of contextual features: the observation features extracted from the neighboring tokens and the labels assigned to those tokens. We call the first one "contextual observation features" and the second "contextual label features". Since in reference parsing, structural SVM is implemented as a sequence learning algorithm, the joint feature presentation function $\psi(x, y)$ includes two kinds of features: state transition features and observation features extracted from individual tokens within a sequence. State transition features utilize contextual label information to model the dependencies between adjacent labels. Having these similar types of feature representations as Hidden Markov Models, structural SVM designed specifically for sequence labeling is sometimes called SVM$^{\mathrm{HMM}}$. In addition to contextual label features, we also combine contextual observation features from neighboring tokens for sequence classification.

Results and discussion

We randomly selected 600 references for training and 1800 references for testing from 1000 HTML articles collected from the top 100 journals cited in the MEDLINE 2006 database. We manually labeled these 2400 references. There are 18003 words in the training references and 53622 words in the testing references. Each entity in reference parsing is a single word, also called a token. The algorithm performance is evaluated at two levels. One is at token-level, i.e., the accuracy of labeling individual tokens. The other is at chunk-level, i.e., the percentage of the entity chunks correctly identified, where an entity *chunk* is the set of consecutive words

Table 2 Features extracted from each token in a reference

1.Author Name Feature	Is the word in Author Name dictionary?
2. Article Title Feature	Is the word in Article Title dictionary?
3. Journal Title Feature	Is the word in Journal Title dictionary?
4. Pagination Pattern	Is the word in pagination formation, e.g., 200-5, H100-H105?
5. Name Initial Pattern	Is the word in name initial pattern, e.g., J.Z., J.-Z.?
6. Four Digit Year Pattern	Is the word in four digit year pattern, e.g., 2005? It must be not before 1500, and not later than the current year.
7. et, al	Is the word "et" or "al", or "et.", or "al."?
8. pp., p.	Is the word "pp.", or "p.", or "pp", or "p"?
9. Ended With "."	Does the word end with "."?
10. Upper Case First Char	Is the first character of the word upper case?
11. Letter Only	Does the word contain letters only?
12. Digit Only	Does the word contain digits only?
13. Digit and Letter	Does the word contain both digits and letters?
14. Digit and Letter Only	Does the word contain digits and letters only?
15. Normalized position	The position of the word normalized by the total number of words in the reference.

having the same entity label. For example, in Table 1 (e), the Citation Number chunk is a single word "12" and the Author chunk is "T.J. McCarthy et al." consisting of four words. The total number of words and chunks for each of the 8 entities in testing references are shown in Table 3. The number of words for Citation Number (742) is larger than the number of chunks (627) is due to the existence of author-year style Citation Numbers, which have more than one word.

Evaluation of structural SVM

For our experiments, we use the SVM^{HMM} library, an implementation of structural SVM for sequence labeling [22], and the linear-kernel since other kernels, e.g. radial basic function (RBF), can be extremely computation intensive. To compare this with SVM, we use LibSVM [23], a library developed at National Taiwan University for word classification. Here linear kernel function is also adopted to facilitate a fair comparison. All the meta-parameters in both SVM and structural SVM are determined with cross-validation on training samples.

We extract 15 observation features including 14 binary features and one normalized position feature from each token. For both SVM and structural SVM, we use 3 sets of features: observation features from the token itself (15 features), observation features from the token and its two neighbors (45 features), and observation features from the token and its four neighbors (75 features). We call the observation features extracted from the neighboring tokens contextual observation features. Specifically, observation features from the immediate left and right neighbors are named as the first order contextual observation features; observation features from the left two and right two neighbors are referred to the second order contextual observation features, and so on. In structural SVM, contextual labels from neighboring tokens are also utilized to explore the dependencies between adjacent tokens. Tables 4, 5, and 6 show the overall token classification accuracies and chunk-level accuracies obtained by SVM and structural SVM for the extraction of 8 entities from the references.

We first use only the 15 observation features from the token itself. Since SVM does not use contextual features, it provides a baseline performance by analyzing only the token itself. As expected, the performance is relatively low: the token classification accuracy is 93.03% and the overall chunk accuracy is only 79.12%. Although structural SVM does not use contextual observation features, it does use the contextual label features. The overall accuracies at token-level and chunk-level are 98.41% and 95.35%, respectively, which are much better than those of the SVM method. This clearly indicates the value of contextual label features in structural SVM.

Table 4 Token classification accuracy obtained by SVM and structural SVM

	SVM	Structural SVM
Features from token itself (15 features)	93.03%	98.41%
Features from the token and its two neighbors (45)	98.20%	98.91%
Features from the token and its four neighbors (75)	98.65%	99.02%

We then add the observation features from the immediate left and right neighbors (the first order contextual observation features). The corresponding token classification accuracy and overall chunk-level accuracy of SVM significantly increase to 98.20% and 94.27%. This indicates that the first order contextual observation features are very important for SVM classification. After combining observation features from one further left and one further right neighbors, the corresponding token-level and chunk-level accuracies increase to 98.65% and 95.59%. This indicates that the second order contextual observation features are still helpful, but less so than the first order ones. For the structural SVM method, when the first order contextual observation features are added, the overall accuracies at token-level and chunk-level increase to 98.91% and 96.81%, respectively. The accuracy improvement is not so substantial as that compared to the SVM method, which may imply that the contextual observation features and contextual label features share redundant discriminative information. After including the second order contextual observation features, there is virtually no performance gain for the structural SVM method, even though it uses extra contextual label features.

Comparison with Conditional Random Field (CRF)

We also compared our methods to CRF, another state-of-the-art sequence learning algorithm [24]. Because only binary features can be used in CRF models, we removed the normalized position feature from the feature vector used in previous evaluation. We then repeated some experiments using the same set of binary features in SVM, structural SVM and CRF methods for

Table 3 The total number of words and chunks for each of the 8 entities in references for evaluation

	Citation Number	Author	Title	Journal	Volume	Year	Pagination	Other	Overall
Total number of words	742	18273	16346	4608	1739	1791	2106	8017	53622
Total number of chunks	627	1800	1308	1758	1735	1791	1751	1708	12478

Table 5 Chunk-level accuracies of SVM method

	Citation Number	Author	Title	Journal	Volume	Year	Pagination	Other	Overall
Features from token itself	93.47%	74.28%	41.90%	51.82%	94.52%	99.50%	93.95%	83.37%	79.12%
Features from the token and its two neighbors	98.73%	92.78%	81.04%	89.48%	99.25%	99.83%	98.63%	93.91%	94.27%
Features from the token and its four neighbors	98.73%	95.11%	84.33%	92.61%	99.31%	99.83%	98.91%	94.91%	95.59%

Table 6 Chunk-level accuracies of structural SVM method

	Citation Number	Author	Title	Journal	Volume	Year	Pagination	Other	Overall
Features from token itself	99.04%	98.94%	78.59%	91.24%	98.90%	99.50%	98.63%	95.90%	95.35%
Features from the token and its two neighbors	99.04%	96.39%	90.60%	94.31%	99.14%	99.94%	98.74%	96.08%	96.81%
Features from the token and its four neighbors	99.20%	97.17%	90.29%	94.94%	99.14%	99.83%	98.63%	95.84%	96.95%

Table 7 Token classification accuracy obtained by SVM, structural SVM and CRF

	SVM	Structural SVM	CRF
Features from the token and its four neighbors (70 features)	97.84%	98.99%	98.91%

fair comparisons. We use SimpleTagger, a sequence tagging tool for CRF implementation in MALLET [25] for our CRF experiments.

We conducted the experiments by extracting 14 binary features plus the second order contextual features, for a total of 70 features from each token. The accuracies obtained at token-level and chunk-level by SVM, structural SVM and CRF are shown in Tables 7 and 8. Compared to the numbers in Tables 4, 5 and 6, the accuracies for both SVM and structural SVM drop a little due to the absence of the normalized position feature. Structural SVM achieved 98.99% token classification accuracy, higher than those of SVM (97.84%) and CRF (98.91%). However, CRF obtained 96.93% overall chunk-level accuracy, higher than that of structural SVM. Since both structural SVM and CRF are sequence learning methods, we do observe that they achieve overall higher token- and chunk-level accuracies than SVM in reference parsing.

The accuracies in CRF experiments are a little different from those reported in [16]. That is because in [16], additional large number of word features is extracted from each token and used in the classification. Adding those word features significantly increases the feature dimensionality, which causes difficulties in training SVM and structural SVM. On the other hand, adding those thousands of word features in CRF improves accuracy only slightly, indicating the non-importance of word features. Basically, we use the first 14 binary features described in Table 2 for a fair comparison.

Conclusions

We have compared SVM and structural SVM as methods for parsing references that appear in medical journal articles. One important difference between the two methods is that the SVM uses only the contextual observation features, while structural SVM uses these as well as contextual label features. Although SVM performance improves greatly and is close to that of structural SVM when the second order contextual observation features are used, structural SVM achieves higher overall token-level and chunk-level accuracies than the SVM method. Both methods achieve above 98% token classification accuracy and an overall chunk-level accuracy of over 95%. Compared to the CRF, we find that the structural SVM achieves similar performance. However, both methods perform better than SVM, showing the advantage of their stronger sequence learning ability.

Reference parsing is considered a sequence learning problem due to the strong regular internal structure in each reference. Additionally, we note that references cited in any one article generally follow the same style. Further exploiting this consistency in consecutive references to improve the performance of reference parsing will be the subject of future work.

Table 8 Chunk-level accuracies of SVM, structural SVM and CRF

	Citation Number	Author	Title	Journal	Volume	Year	Pagination	Other	Overall
SVM	99.04%	93.06%	78.44%	92.38%	98.85%	99.78%	98.74%	93.03%	94.29%
Structural SVM	98.89%	96.39%	90.21%	94.99%	99.25%	99.83%	98.80%	95.78%	96.82%
CRF	98.57%	97.83%	90.75%	94.99%	98.96%	99.22%	98.91%	95.61%	96.93%

Acknowledgements

This research was supported by the Intramural Research Program of the National Institutes of Health (NIH), National Library of Medicine, and Lister Hill National Center for Biomedical Communications.

This article has been published as part of *BMC Bioinformatics* Volume 12 Supplement 3, 2011: Machine Learning for Biomedical Literature Analysis and Text Retrieval. The full contents of the supplement are available online at http://www.biomedcentral.com/1471-2105/12?issue=S3.

Authors' contributions

XZ implemented the algorithms, drafted the manuscript and is the corresponding author. JZ participated in drafting the manuscript and provided the previous work for reference parsing. DXL and GRT supervised the research and revised the manuscript. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Published: 9 June 2011

References

1. Lawrence S, Giles CL, Bollacker K: **Digital libraries and autonomous citation indexing.** *IEEE Computer* 1999, **vol. 32, 6**:67-71.
2. **ISI Web of Knowledge.** [http://www.isiwebofknowledge.com/].
3. **Google Scholar.** [http://scholar.google.com/].
4. Lee D, Kang J, Mitra P, Giles CL, On BW: **Are your citations clean?** *Communications of the ACM* 2007, **50(12)**:33-38.
5. Kim I, Le DX, Thoma GR: **Identification of "comment-on sentences" in online biomedical documents using support vector machines.** *Proc. of SPIE conference on Document Recognition and Retrieval* 2007, **68150X**:1-9.
6. Aronson AR, Bodenreider O, Chang HF, Humphrey SM, Mork JG, Nelson SJ, Rindflesch TC, Wilbur WJ: **The NLM indexing initiative.** *Proc. of AMIA Symp* 2000, 17-21.
7. Chowdhury G: **Template mining for information extraction from digital documents.** *Library Trends* 1999, **48(1)**:182-208.
8. Ding Y, Chowdhury G, Foo S: **Template mining for the extraction of citation from digital documents.** *Proc. of the 2nd Asian Digital Library Conference* 1999, 47-62.
9. Day MY, Tsai TH, Sung CL, Lee CW, Wu SH, Ong CS, Hsu WL: **A knowledge-based approach to citation extraction.** *IEEE Int'l Conf. on Information Reuse and Integration* 2005, 50-55.
10. Day MY, Tsai TH, Sung CL, Hsieh CC, Lee CW, Wu SH, Wu KP, Ong CS, Hsu WL: **Reference metadata extraction using a hierarchical knowledge representation framework.** *Decision Support Systems* 2007, **43(1)**:152-167.
11. Huang IA, Ho JM, Kao HY, Lin WC: **Extracting citation metadata from online publication lists using BLAST.** *Proc. of the Eighth Pacific-Asia Conference on Knowledge Discovery and Data Mining* 2004, 26-28.
12. Parmentier F, Belaïd A: **Logical structure recognition of scientific bibliographic references.** *Proc. of ICDAR* 1997, **2**:1072-1076.
13. Besagni D, Belaïd A, Benet N: **A segmentation method for bibliographic references by contextual tagging of fields.** *Proc. of ICDAR* 2003, **1**:384-388.
14. Takasu A: **Bibliographic attribute extraction from erroneous references based on a statistical model.** *Proc. of JCDL* 2003, 49-60.
15. Okada T, Takasu A, Adachi J: **Bibliographic component extraction using support vector machines and Hidden Markov Models.** *Proc. of ECDL* 2004, 501-512.
16. Zou J, Le DX, Thoma GR: **Locating and parsing bibliographical references in the HTML medical journal articles.** *International Journal on Document Analysis and Recognition* 2010, **13(2)**:107-119.
17. Cortez E, da Silva AS, Goncalves MA, Mesquita F, de Moura ES: **A flexible approach for extracting metadata from bibliographic citations.** *Journal of the American Society for Information Science and Technology* 2009, **60(6)**:1144-1158.
18. Councill IG, Giles CL, Kan KY: **ParsCit: an open-source CRF reference string parsing package.** *Proc. of the Language Resources and Evaluation Conference (LREC08)* 2008 [http://wing.comp.nus.edu.sg/parsCit/].
19. **FreeCite.** [http://freecite.library.brown.edu/welcome].
20. Tsochantaridis I, Hofmann T, Joachims T, Altun Y: **Support vector machine learning for interdependent and structured output spaces.** *Int'l Conf. on Machine Learning(ICML)* 2004, 104-112.
21. Joachims T, Finley T, Yu CN: **Cutting-plane training of structural SVMs.** *Machine Learning Journal* 2009, **77(1)**:27-59.
22. Herbst E, Joachims T: **SVMHMM: sequence tagging with structural support vector machine.** 2008 [http://www.cs.cornell.edu/People/tj/svm_light/svm_hmm.html].
23. Chang CC, Lin CJ: **LIBSVM: a library for support vector machines.** 2001, Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.
24. Lafferty J, McCallum A, Pereira F: **Conditional random fields: probabilistic models for segmenting and labeling sequence data.** *Proc. of ICML* 2010, 282-289.
25. McCallum AK: **MALLET: a machine learning for language toolkit.** 2002 [http://mallet.cs.umass.edu/index.php].

doi:10.1186/1471-2105-12-S3-S7
Cite this article as: Zhang *et al.*: A structural SVM approach for reference parsing. *BMC Bioinformatics* 2011 **12**(Suppl 3):S7.