Available online at www.sciencedirect.com
SCIENCE DIRECT®
Information Processing and Management 42 (2006) 963–979
INFORMATION PROCESSING & MANAGEMENT
www.elsevier.com/locate/infoproman

Information extraction from research papers using conditional random fields ☆

Fuchun Peng [a,*], Andrew McCallum [b]

[a] *BBN Technologies, 50 Moulton Street, Cambridge, MA 02138, United States*
[b] *Department of Computer Science, University of Massachusetts Amherst, 140 Governors Drive, Amherst, MA 01003, United States*

Received 31 March 2005
Available online 6 December 2005

Abstract

With the increasing use of research paper search engines, such as CiteSeer, for both literature search and hiring decisions, the accuracy of such systems is of paramount importance. This article employs conditional random fields (CRFs) for the task of extracting various common fields from the headers and citation of research papers. CRFs provide a principled way for incorporating various local features, external lexicon features and globle layout features. The basic theory of CRFs is becoming well-understood, but best-practices for applying them to real-world data requires additional exploration. We make an empirical exploration of several factors, including variations on Gaussian, Laplace and hyperbolic-L_1 priors for improved regularization, and several classes of features. Based on CRFs, we further present a novel approach for constraint co-reference information extraction; i.e., improving extraction performance given that we know some citations refer to the same publication. On a standard benchmark dataset, we achieve new state-of-the-art performance, reducing error in average F1 by 36%, and word error rate by 78% in comparison with the previous best SVM results. Accuracy compares even more favorably against HMMs. On four co-reference IE datasets, our system significantly improves extraction performance, with an error rate reduction of 6–14%.

Keywords: Information extraction; Constraint information extraction; Conditional random fields; Regularization

1. Introduction

Research paper search engines, such as *CiteSeer* (Lawrence, Giles, & Bollacker, 1999) and *Cora* (McCallum, Nigam, Rennie, & Seymore, 2000), give researchers tremendous power and convenience in their research. They are also becoming increasingly used for recruiting and hiring decisions. Thus the information quality of

☆ This work was mostly conducted while the first author was at the University of Massachusetts Amherst.
* Corresponding author.
E-mail addresses: fpeng@bbn.com, fuchun@cs.umass.edu (F. Peng), mccallum@cs.umass.edu (A. McCallum).

0306-4573/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ipm.2005.09.002

such systems is of significant importance. This quality critically depends on an information extraction component that extracts meta-data, such as title, author, institution, etc., from paper headers and references, because these meta-data are further used in many component applications such as field-based search, author analysis, and citation analysis.

Previous work in information extraction from research papers has been based on two major machine learning techniques. The first is hidden Markov models (HMM) (Seymore et al., 1999; Takasu, 2003). An HMM learns a generative model over input sequence and labeled sequence pairs. While enjoying wide historical success, standard HMM models have difficulty modeling multiple non-independent features of the observation sequence. The second technique is based on discriminatively-trained SVM classifiers (Han et al., 2003). These SVM classifiers can handle many non-independent features. However, for this sequence labeling problem, Han et al. (2003) work in a two stags process: first classifying each line (one reference can span multiple lines) independently to assign it label, then adjusting these labels based on an additional classifier that examines larger windows of labels. Solving the information extraction problem in two steps loses the tight interaction between state transitions and observations.

In this article, we present results on the research paper meta-data extraction task using a Conditional Random Field (Lafferty, McCallum, & Pereira, 2001), and explore several practical issues in applying CRFs to information extraction in general. The CRF approach draws together the advantages of both finite state HMM and discriminative SVM techniques by allowing use of arbitrary, dependent features and joint inference over entire sequences. CRFs have been previously applied to other tasks such as name entity extraction (McCallum & Li, 2003), table extraction (Pinto, McCallum, Wei, & Croft, 2003) and shallow parsing (Sha & Pereira, 2003). The basic theory of CRFs is now well-understood, but the best-practices for applying them to new, real-world data is still in an early-exploration phase. Here we explore two key practical issues: (1) regularization, with an empirical study of Gaussian (Chen & Rosenfeld, 2000), exponential (Goodman, 2003), and hyperbolic-L_1 (Pinto et al., 2003) priors; (2) exploration of various families of features, including text, lexicons, and layout.

One problem in citation domain is that a publication can be cited in various ways. For example, some citations use complete information including author, title, book title, conference venue, date, location, and publisher, while others use only a subset of the fields; Some citations use full author names, while others use only abbreviations; some citations put first name before last name, while others reverse name order. Table 1 shows two different citations referring to the same publication. The two citations have different author formats; citation 2 is more complete: it has full *author* name, full *publisher* name, and it has a *pages* field. Citation 1 uses abbreviation for *author* name, short name for *publisher*, and it does not have a *pages* field. In our datasets, a paper can have up to 21 variant citations. For a paper search engine, it is neither effective nor efficient to store all these variant citations for the same publication. It is desirable to create a canonical citation for such variant citations. When extracting fields from these citations, traditionally they are considered independent of each other (Han et al., 2003; Lawrence et al., 1999). However, such co-referential information could strongly constrain each other in making consistent segmentation decisions. We refer to such a task as constraint information extraction.

Our second contribution in this article is to address the constraint information extraction issue. We propose a novel approach for creating canonical citations for a publication and improving citation segmentation. Given a number of co-referential citations referring to the same publication, our model creates a canonical citation for this publication based on segmentation uncertainty, and then uses the created canonical citation to re-rank the segmentations for each individual citation.

In standard segmentation experiments, we describe a large collection of experimental results on a number of real world datasets. Dramatic improvements are obtained in comparison with previous SVM and HMM based results, reducing average F1 error by 36%, and word error rate by 78% in comparison with the previous best

Table 1
Two different citations for the same publication

1: B. Laurel, Interface Agents: Metaphors with Character, in The Art of Human Computer Interface Design, B. Laurel (ed), AddisonWesley, 1990

2: Brenda Laurel. "Interface Agents: Metaphors with Character." In The Art of Human-Computer Interface Design, pages 355–365. Addison-Wesley Publishing Company, 1990

SVM results. In co-referent constraint information extraction, we present experimental results on the four sections of CiteSeer citation-matching data (Lawrence et al., 1999), with a significant error rate reduction of 6–14% on extraction performance.

We organize the rest of the paper as follows. We first describe conditional random fields in Section 2. Then we explore the regularization issue and feature engineering issues in Sections 3 and 4. In Section 5 we propose a novel approach for co-reference constraint information extraction. We systematically study the these problems in Section 6 and conclude the paper in Section 7.

2. Conditional random fields

Conditional random fields (CRFs) are undirected graphical models trained to maximize a conditional probability (Lafferty et al., 2001). A common special-case graph structure is a linear chain as depicted in Fig. 1, which corresponds to a finite state machine, and is suitable for sequence labeling. A linear-chain CRF with parameters $\Lambda = \{\lambda, \ldots\}$ defines a conditional probability for a state (or label[1]) sequence $y = y_1 \cdots y_T$ given an input sequence $x = x_1 \cdots x_T$ to be

$$P_\Lambda(y|x) = \frac{1}{Z_x} \exp\left(\sum_{t=1}^{T} \sum_k \lambda_k f_k(y_{t-1}, y_t, x, t)\right), \tag{1}$$

where Z_x is the normalization constant that makes the probability of all state sequences sum to one, $f_k(y_{t-1}, y_t, x, t)$ is a feature function which is often binary-valued, but can be real-valued, and λ_k is a learned weight associated with feature f_k. The feature functions can measure any aspect of a state transition, $y_{t-1} \to y_t$ and the observation sequence, x, centered at the current time step, t. For example, one feature function might have value 1 when y_{t-1} is the state TITLE, y_t is the state AUTHOR, and x_t is a word appearing in a lexicon of people's first names. Large positive values for λ_k indicate a preference for such an event, while large negative values make the event unlikely.

Given such a model as defined in Eq. (1), the most probable labeling sequence for an input x,

$$y^* = \arg\max_y P_\Lambda(y|x),$$

can be efficiently calculated by dynamic programming using the Viterbi algorithm. Calculating the marginal probability of states or transitions at each position in the sequence by a dynamic-programming-based inference procedure is very similar to forward–backward for hidden Markov models.

The parameters may be estimated by maximum likelihood—maximizing the conditional probability of a set of label sequences, each given their corresponding input sequences. The log-likelihood of training set $\{(x_i, y_i): i = 1, \ldots, M\}$ is written

$$L_\Lambda = \sum_i \log P_\Lambda(y_i|x_i) = \sum_i \left(\sum_{t=1}^{T} \sum_k \lambda_k f_k(y_{t-1}, y_t, x, t) - \log Z_{x_i}\right). \tag{2}$$

Following the standard exponential family model theory, we maximize Eq. (2), which corresponds to satisfying the following equality, wherein the empirical count of each feature matches its expected count according to the model $P_\Lambda(y|x)$.

$$\sum_i \sum_i f_k(y_{t-1}, y_t, x_i, t) = \sum_i \sum_{y'} P_\Lambda(y'|x_i) \sum_t f_k(y'_{t-1}, y'_t, x_i, t).$$

CRFs share all of the advantageous properties of standard maximum entropy models, including their convex likelihood function, which guarantees that the learning procedure converges to the global maximum. Traditional maximum entropy learning algorithms, such as GIS and IIS (Pietra, Pietra, & Lafferty, 1995), can be used to train CRFs, however, it has been found that a quasi-Newton gradient-climber, BFGS, converges much

[1] We consider here only finite state models in which there is a one-to-one correspondence between states and labels; this is not, however, strictly necessary.

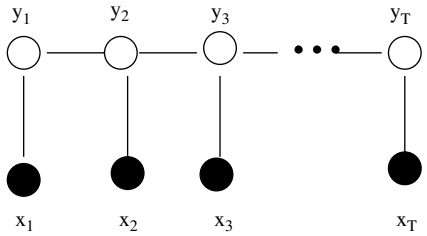

Fig. 1. Graphical model for linear chain structure CRF.

faster (Malouf, 2002; Sha & Pereira, 2003). We use BFGS for optimization. In our experiments, we shall focus instead on two other aspects of CRF deployment, namely regularization and selection of different model structures and feature types.

3. Regularization in CRFs

To avoid over-fitting, log-likelihood is often penalized by some prior distribution over the parameters. To get an idea how the parameters are distributed, we trained a CRF with Gaussian prior on one of our datasets and graphed the distribution of these parameters. Fig. 2 shows an empirical distribution of parameters, Λ. Interestingly, the graph has a spike at zero. Three prior distributions that have a shape similar to this empirical distribution are the Gaussian prior, exponential prior, and hyperbolic-L_1 prior, each shown in Fig. 3. In this paper we provide an empirical study of these three priors.

3.1. Gaussian prior

With a Gaussian prior, log-likelihood (2) is penalized as follows:

$$L_\Lambda = \sum_i \log P_\Lambda(y_i|x_i) - \sum_k \frac{\lambda_k^2}{2\sigma_k^2}, \tag{3}$$

where σ_k^2 is a variance.

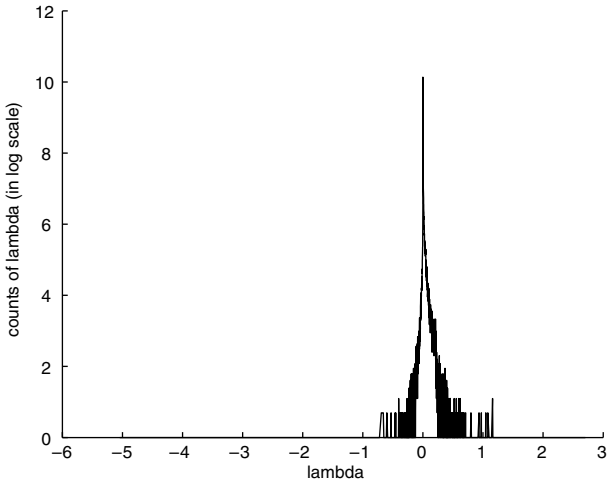

Fig. 2. Empirical distribution of λ.

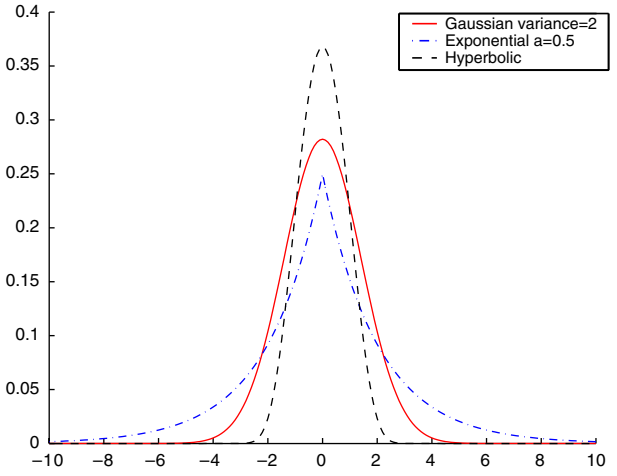

Fig. 3. Shapes of prior distributions.

Maximizing (3) corresponds to satisfying

$$\sum_i \sum_t f_k(y_{t-1}, y_t, x_i, t) - \frac{\lambda_k}{\sigma_k^2} = \sum_i \sum_{y'} P_\Lambda(y'|x_i) \sum_t f_k(y'_{t-1}, y'_t, x_i, t).$$

This adjusted constraint (as well as the adjustments imposed by the other two priors) is intuitively understandable: rather than matching exact empirical feature frequencies, the model is tuned to match discounted feature frequencies. Chen and Rosenfeld (2000) discuss this in the context of other discounting procedures common in language modeling. We call the term subtracted from the empirical counts (in this case λ_k/σ^2) *a discounted value.*

The variance can be feature dependent. However for simplicity, constant variance is often used for all features. In this paper, however, we experiment with several alternate versions of Gaussian prior in which the variance is feature dependent.

Although Gaussian (and other) priors are gradually overcome by increasing amounts of training data, perhaps not at the right rate. The three methods below all provide ways to alter this rate by changing the variance of the Gaussian prior dependent on feature counts.

1. Threshold cut: In language modeling, e.g., Good-Turing smoothing, only low frequency words are smoothed. Here we apply the same idea and only smooth those features whose frequencies are lower than a threshold (seven in our experiments, following standard practice in language modeling).
2. Divide count: Here we let the discounted value for a feature depend on its frequency in the training set, $c_k = \sum_i \sum_t f_k(y_{t-1}, y_t, x, t)$. The discounted value used here is $\frac{\lambda_k}{c_k \times \sigma^2}$ where σ is a constant over all features. In this way, we increase the smoothing on the low frequency features more so than the high frequency features.
3. Bin-based: We divide features into classes based on frequency. We bin features by frequency in the training set, and let the features in the same bin share the same variance. The discounted value is set to be $\frac{\lambda_k}{\lceil c_k/N \rceil \times \sigma^2}$ where c_k is the count of features, N is the bin size, and $\lceil a \rceil$ is the ceiling function. Alternatively, the variance in each bin may be set independently by cross-validation.

3.2. Exponential prior

Whereas the Gaussian prior penalizes according to the square of the weights (an L_2 penalizer), the intention here is to create a smoothly differentiable analogue to penalizing the absolute-value of the weights (an L_1 penalizer). L_1 penalizers often result in more "sparse solutions", in which many features have weight nearly at zero, and thus provide a kind of soft feature selection that improves generalization.

Goodman (2003) proposes an exponential prior, specifically a Laplacian prior, as an alternative to Gaussian prior. Under this prior,

$$L_\Lambda = \sum_i \log P_\Lambda(y_i|x_i) - \sum_k \alpha_k|\lambda_k|, \tag{4}$$

where α_k is a parameter in exponential distribution.

Maximizing (4) would satisfy

$$\sum_i \sum_t f_k(y_{t-1}, y_t, x_i, t) - \text{sign}(\lambda_k) * \alpha_k = \sum_i \sum_{y'} P_\Lambda(y'|x_i) \sum_t f_k(y'_{t-1}, y'_t, x_i, t),$$

where $\text{sign}(\lambda_k)$ is the sign of λ_k.

This corresponds to the absolute smoothing method in language modeling. We set the $\alpha_k = \alpha$; i.e. all feature share the same constant whose value can be determined using absolute discounting $\alpha = \frac{n_1}{n_1+2n_2}$, where n_1 and n_2 are the number of features occurring once and twice (Ney, Essen, & Kneser, 1995).

3.3. Hyperbolic-L_1 prior

Another L_1 penalizer is the hyperbolic-L_1 prior, described in Pinto et al. (2003). The hyperbolic distribution has log-linear tails. Consequently the class of hyperbolic distributions is an important alternative to the class of normal distributions. While less frequently used in natural language processing, it has been commonly used for analyzing data from various scientific areas such as finance.

Under a hyperbolic prior,

$$L_\Lambda = \sum_i \log P_\Lambda(y_i|x_i) - \sum_k \log\left(\frac{e^{\lambda_k} + e^{-\lambda_k}}{2}\right) \tag{5}$$

which corresponds to satisfying

$$\sum_i \sum_t f_k(y_{t-1}, y_t, x_i, t) - \frac{e^{\lambda_k} - e^{-\lambda_k}}{e^{\lambda_k} + e^{-\lambda_k}} = \sum_i \sum_{y'} P_\Lambda(y'|x_i) \sum_i f_k(y'_{t-1}, y'_t, x_i, t).$$

The hyperbolic prior was also tested with CRFs in McCallum and Li (2003).

4. Exploration of feature space

Wise choice of features is always vital to the performance of any machine learning solution. Feature induction (McCallum, 2003) has been shown to provide significant improvements in CRFs performance. The focus in this section is on three other aspects of the feature space.

4.1. State transition features

In CRFs, state transitions are also represented as features. The feature function $f_k(y_{t-1}, y_t, x, t)$ in Eq. (1) is a general function over states and observations. Different state transition features can be defined to form different Markov-order structures. We define four different state transitions features corresponding to different Markov order for different classes of features. Higher-order features model dependencies better, but also create more data sparse problem and require more memory in training.

1. First-order: Here the inputs are examined in the context of the current state only. The feature functions are represented as $f(y_t, x)$. There are no separate parameters or preferences for state transitions at all.
2. First-order + transitions: Here we add parameters corresponding to state transitions. The feature functions used are $f(y_t, x)$, $f(y_{t-1}, y_t)$.
3. Second-order: Here inputs are examined in the context of the current and previous states. Feature function are represented as $f(y_{t-1}, y_t, x)$.

Table 2
List of used features

Feature name	Description
Local features	
INITCAP	Starts with a capitalized letter
ALLCAPS	All characters are capitalized
CONTAINSDIGITS	Contains at least one digit
ALLDIGITS	All characters are digits
PHONEORZIP	Phone number or zip code
CONTAINSDOTS	Contains at least one dot
CONTAINSDASH	Contains at least one -
ACRO	Acronym
LONELYINITIAL	Initials such as *A*.
SINGLECHAR	One character only
CAPLETTER	One capitalized character
PUNC	Punctuations
URL	URLs
EMAIL	Emails
WORD	Word itself
Layout features	
LINE_START	At the beginning of a line
LINE_IN	In-between a line
LINE_END	At the end of a line
FONT_CHANGE	Font is different from the previous word
External lexicon features	
BIBTEX_AUTHOR	Match an author in the dictionary
BIBTEX_DATE	Words like Jan. Feb.
NOTES	Words like *appeared, submitted*
AFFILIATION	Words like *institution, Labs*, etc.

4. Third-order: Here inputs are examined in the context of the current, two previous states. Feature function are represented as $f(y_{t-2}, y_{t-1}, y_t, x)$.

4.2. Local features, layout features and lexicon features

One of the advantages of CRFs (and also standard maximum entropy models in general) is that they easily afford the use of arbitrary features of the input. One can encode local spelling features, layout features such as positions of line breaks and font features, as well as external lexicon features, all in one framework. We study all these features in our research paper extraction problem, evaluate their individual contributions, and give some guidelines for selecting good features.

To analyze the contribution of different kinds of features, we divide the features to be local features, layout features, and external lexicon resources. The features are summarized in Table 2.

5. Co-reference constraint information extraction

In the citation domain, a publication can be cited in different ways. These variant citations are co-referent. Traditionally, when segmenting these citations, they are considered independent of each other. However, intuitively it should be beneficial to consider them simultaneously since the co-reference information strongly constraints and imposes the consistency of segmentation decisions. Further, it is desirable to create a canonical citation for these variants in order to eliminate duplication and produce uniform citation. To address these problems, we propose a constraint information extraction algorithm which takes advantages of segmentation uncertainty to create canonical citations, which in turn are used to re-rank citation segmentations.

Our solution to co-referent information extraction is based on segmentation uncertainty. In finite state machine based models such as CRFs (Lafferty et al., 2001), Viterbi decoding, which returns only the most likely state sequence, is normally used for segmenting sequences. However, in practice, the most likely sequence may not be the best segmentation. A better segmentation could have lower likelihood, as we will show in experiments. N-best decoding is a technique commonly used in speech recognition (Schwartz & Chow, 1990). It returns the top N-best segmentations ranked by likelihood. One can then use additional information to re-rank the segmentations. The additional information we have here is the knowledge that these citations are referring to the same publication. The re-ranking approach has been successful in other natural language processing tasks such as parsing (Collins, 2000) and named entity extraction (Collins, 2002).

Table 3 is an example of using a canonical citation to re-rank the segmentations. The fields of the canonical citation are illustrated in the first segment in the table. The most likely segmentation is incorrect in that "Addison Wesley" was tagged as a *location* (bolded in the table). The second likely segmentation (shown at the bottom), however is correct. Given the canonical citation, this segmentation would be preferred since it has smaller distance.

Our constraint information extraction model consists of two components. The first component creates a canonical citation by selecting field attributes from the N-best segmentations. The second component uses the created canonical citation to re-rank the segmentations, based on the criteria that a better segmentation gives higher co-referential score.

5.1. Creating a canonical citation

We first clarify several terms. A *publication* is a paper published in some venue. A *citation* is a reference to a publication. A *canonical citation* (also called publication prototype) is a normalized citation which consists of a number canonical database fields for this publication.

Canonical database fields creation (also called data integration) is a challenging problem that has attracted significant research in the areas of artificial intelligence (Basu, Hirsh, Cohen, & Neville-Manning, 2001; Cohen, McAllester, & Kautz, 2000). We create canonical database fields based on segmentation uncertainty. To be more specific, we create a canonical citation with K fields from M co-referent citations; each of the citations has N most likely segmentations. For each of the K fields, we assign its value by selecting an attribute from the corresponding field of the $M \times N$ segmentations. Thus, to create a canonical citation, we have to enumerate all $(M \times N)^K$ possibilities to select the best combination. Fortunately, in practice, there are few enough combinations that exact inference can be performed here. In our experiments, there are a maximum of 13 citation fields (although typically many fewer), 2–21 citations (typically less than 10) in a co-referent cluster, and N was 5 or less. However, when M becomes large, one must resort to approximation algorithms to compute the prototype.

Given the attributes of a publication, we compute scores for all (*publication, segmentation*) pairs. The segmentation with the highest score is selected as the best segmentation for the citation. These publication attributes are also scored by summing these highest scores for all citations. In the end, the publication attributes with the highest score are chosen as the canonical citation for this publication.

Table 3
An example of a prototype (first segment) and two alternative segmentations for a single citation

Authors: B. Laurel
Pages: pages 355–365
Title: Interface Agents: Metaphors with Character
Publisher: Addison Wesley
Date: 1990
Editor: B. Laurel (ed)
⟨author⟩ B. Laurel, ⟨/author⟩ ⟨title⟩ Interface Agents: Metaphors with Character, ⟨/title⟩ ⟨booktitle⟩ in The Art of Human Computer Interface Design, ⟨/booktitle⟩ ⟨editor⟩ B. Laurel (ed), ⟨/editor⟩ ⟨**location**⟩ Addison Wesley, ⟨/**location**⟩ ⟨date⟩ 1990. ⟨/date⟩
⟨author⟩ Laurel, B., ⟨/author⟩ ⟨title⟩ Interface Agents: Metaphors with Character, ⟨/title⟩ ⟨booktitle⟩ in The Art of Human Computer Interface Design, ⟨/booktitle⟩ ⟨editor⟩ B. Laurel (ed), ⟨/editor⟩ ⟨**publisher**⟩ AddisonWesley, ⟨/**publisher**⟩ ⟨date⟩ 1990. ⟨/date⟩

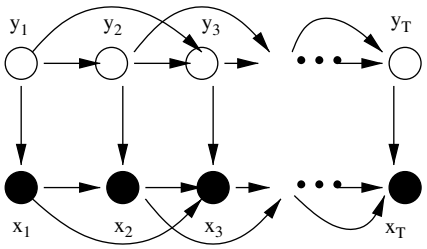

Fig. 4. Higher-order hidden Markov model with observation dependency.

5.2. Improving segmentation based on canonical citation

After a canonical citation is created, we can use it to re-rank the N-best segmentations for each individual citation. The rational for re-ranking is that a better segmentation gives better co-referential score with the canonical citation. The co-referential score between the canonical citation and a segmentation is measured by a field string distance, which is the sum of the edited distance of all fields.[2] A segmentation with shorter distance to the prototype is considered as a better segmentation.

6. Empirical study

6.1. Hidden Markov models

Here we also briefly describe a HMM model we used in our experiments as baselines. We relax the independence assumption made in standard HMM and allow Markov dependencies among observations as depicted in Fig. 4. As in CRFs, we can vary Markov orders in state transition and observation transitions. In our experiments, a model with second-order state transitions and first-order observation transitions performs the best. The state transition probabilities and emission probabilities are estimated using maximum likelihood estimation with absolute smoothing.

6.2. Datasets

In standard information extraction experiments, we experiment with three datasets of research paper content. Two consist of the headers of research papers. The other consists of pre-segmented citations from the reference sections of research papers. The citation dataset and one of the header datasets are part of a standard benchmark on which previous studies have evaluated performance (Han et al., 2003; McCallum et al., 2000; Seymore et al., 1999). The other header dataset was created by the authors due to inadequacies with the standard benchmark (no font or formatting information).

However, these datasets do not have co-referent citations and we have to use other datasets for our co-reference information extraction experiments. We use CiteSeer data (Lawrence et al., 1999) as our co-reference information extraction dataset.

6.2.1. Paper header dataset

A header of a research paper is defined to be all of the words from the beginning of the paper up to either the first section of the paper, usually the introduction, or to the end of the first page, whichever occurs first. It contains 15 fields to be extracted: title, author, affiliation, address, note, email, date, abstract, introduction, phone number, keywords, web address, degree, publication number, and page (Seymore et al., 1999).

The first header dataset contains 935 headers. Following previous research (Han et al., 2003; McCallum et al., 2000; Seymore et al., 1999), we use a randomly selected 500 samples for training and the remaining 435 for testing. We refer this dataset as **H1**.

[2] We use the SecondString package (Cohen, 2003).

We have also created a second header dataset, comprising 450 headers. This dataset contains font information which we expected to be an good indicator of field boundaries. The papers are randomly selected among 8000 papers we found by crawling from Internet across many university and research institution websites. We use 300 samples for training and 150 for testing. We refer this dataset as **H2**.

6.2.2. Paper reference dataset

The reference dataset was created by the Cora project (McCallum et al., 2000). It contains 500 references, we use 350 references for training and the remaining 150 for testing. References contain 13 fields: author, title, editor, booktitle, date, journal, volume, tech, institution, pages, location, publisher, note. We refer this dataset as **R1**.

6.2.3. Co-reference IE dataset

The co-reference dataset contains approximately 1500 citations to 900 papers. The citations have been manually labeled for co-reference and manually segmented into fields, such as author, title, etc. The dataset has four subsets of citations, each one centered around a topic (e.g., reinforcement learning). The statistics of the four sections is shown in Table 4.

6.3. Performance measures

To give a comprehensive evaluation, we measure extraction performance using several different metrics. In addition to the previously-used word accuracy measure (which overemphasises accuracy of the *abstract* field), we use per-field F1 measure (both for individual fields and averaged over all fields—called a "macro average" in the information retrieval literature), and whole instance accuracy for measuring overall performance in a way that is sensitive to an error in any part of header or citation.

6.3.1. Measuring field-specific performance

1. Word Accuracy: We define A as the number of true positive words, B as the number of false negative words, C as the number of false positive words, D as the number of true negative words, and $A+B+C+D$ is the total number of words. Word accuracy is calculated to be $\frac{A+D}{A+B+C+D}$.
2. F1-measure: Precision, recall and F1 measure are defined as follows:

$$\text{Precision} = \frac{A}{A+C},$$

$$\text{Recall} = \frac{A}{A+B},$$

$$\text{F1} = \frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}.$$

6.3.2. Measuring overall performance

1. Overall word accuracy: Overall word accuracy is the percentage of words whose predicted labels equal their true labels. Word accuracy favors fields with large number of words, such as the *abstract*.
2. Averaged F-measure: Averaged F-measure is computed by averaging the F1-measures over all fields. Average F-measure favors labels with small number of words, which complements word accuracy. We should look at both word accuracy and average F-measure in evaluation.

Table 4
CiteSeer dataset for co-reference information extraction

	Reinforce	Face	Reasoning	Constraint
Number of citations	406	349	514	295
Number of papers	148	242	296	199

3. Whole instance accuracy: An instance here is defined to be a single header or reference. Whole instance accuracy is the percentage of instances whose words are all correctly labeled.

6.4. Standard information extraction results

We first report the overall results by comparing CRFs with HMMs, and with the previously best benchmark results obtained by SVMs (Han et al., 2003). We then break down the results to analyze various factors individually. Table 5 shows the results on dataset **H1** with the best results in bold; [*intro* and *page* fields are now shown, following past practice (Seymore et al., 1999; Han et al., 2003)]. The results we obtained with CRFs use second-order state transition features and line-break layout features. The results we obtained with the HMM model use a second-order model for transitions, and a first-order for observations. The results on SVM were obtained from Han et al. (2003) by computing F1 measures from the precision and recall numbers they report.

Table 6 shows the results on dataset **H2**. Table 7 shows the results on dataset **R1**. SVM results are not available for these datasets.

6.5. Constraint information extraction results

One hypothesis behind the constraint information extraction model is that the best segmentation (with the highest word accuracy) is not always the most likely one as returned by the Viterbi algorithm. Instead, the best segmentation could be another one from the N-best list. To verify this hypothesis, we conduct an experiment that always selects the segmentation that has the highest word accuracy from the N-best list. This experiment serves as a upper bound performance that our model could achieve.

The upper bound results (F-measure) on each of the four topics in our co-reference dataset are summarized in Table 8. The big gap between top N and top 1 performance indicates great potential to improve segmentation based on optimally selecting segmentations.

The second hypothesis behind our approach is that co-reference information provides extraction with strong constraints and should be beneficial to extraction. To verify this, we conduct co-referent information extraction using true co-reference information contained in our data. Table 9 shows the improved segmentation performance using true co-reference information. We achieve 3.8–17.9% error rate reduction on the four topics of the co-reference dataset. The results reported here only consider citations that were grouped together

Table 5
Extraction results on **H1**

	HMM		CRF		SVM	
Overall acc.	93.1%		**98.3**%		92.9%	
Instance acc.	4.13%		**73.3**%		–	
	Acc.	F1	Acc.	F1	Acc.	F1
Title	98.2	82.2	99.7	**97.1**	98.9	96.5
Author	98.7	81.0	99.8	**97.5**	99.3	97.2
Affiliation	98.3	85.1	99.7	**97.0**	98.1	93.8
Address	99.1	84.8	99.7	**95.8**	99.1	94.7
Note	97.8	81.4	98.8	**91.2**	95.5	81.6
Email	99.9	92.5	99.9	**95.3**	99.6	91.7
Date	99.8	80.6	99.9	**95.0**	99.7	90.2
Abstract	97.1	98.0	99.6	**99.7**	97.5	93.8
Phone	99.8	53.8	99.9	**97.9**	99.9	92.4
Keyword	98.7	40.6	99.7	**88.8**	99.2	88.5
Web	99.9	68.6	99.9	**94.1**	99.9	92.4
Degree	99.5	68.8	99.8	**84.9**	99.5	70.1
Pubnum	99.8	64.2	99.9	86.6	99.9	**89.2**
Average F1		75.6		**93.9**		89.7

Table 6
Extraction results on **H2**

	HMM		CRF	
Overall acc.	93.1%		**98.6**%	
Instance acc.	8%		**74.7**%	
	Acc.	F1	Acc.	F1
Title	97.6	80.9	99.6	**96.9**
Author	98.2	79.1	99.8	**97.7**
Affiliation	97.9	86.4	99.7	**98.1**
Address	98.7	83.6	99.6	**95.2**
Note	98.4	58.6	99.3	**83.2**
Email	99.5	84.0	99.8	**95.5**
Date	99.7	76.7	99.9	**98.9**
Abstract	97.31	98.1	99.8	**99.8**
Phone	99.8	64.7	99.9	**97.8**
Keyword	99.1	18.9	99.8	**91.9**
Web	99.8	67.9	99.9	**93.0**
Degree	99.8	67	100	**100**
Pubnum	99.9	57.1	99.9	**62.5**
Average F1		71.1	97.7	**93.1**

Table 7
Extraction results on paper references on **R1**

	HMM		CRF	
Overall acc.	85.1%		**95.37**%	
Instance acc.	10%		**77.33**%	
	Acc.	F1	Acc.	F1
Author	96.8	92.7	99.9	**99.4**
Booktitle	94.4	0.85	97.7	**93.7**
Date	99.7	96.9	99.8	**98.9**
Editor	98.8	70.8	99.5	**87.7**
Institution	98.5	72.3	99.7	**94.0**
Journal	96.6	67.7	99.1	**91.3**
Location	99.1	81.8	99.3	**87.2**
Note	99.2	50.9	99.7	**80.8**
Pages	98.1	72.9	99.9	**98.6**
Publisher	99.4	**79.2**	99.4	76.1
Tech	98.8	74.9	99.4	**86.7**
Title	92.2	87.2	98.9	**98.3**
Volume	98.6	75.8	99.9	**97.8**
Average F1		77.6		**91.5**

Table 8
Optimal segmentation improvement for different values of N over all citations (including singletons)

	Reinforce	Face	Reason	Constraint
$N = 1$	93.6	91.1	91.2	93.3
$N = 2$	95.0	92.7	92.9	95.7
$N = 3$	95.8	93.7	94.0	96.9
$N = 4$	96.1	94.2	94.3	97.3
$N = 5$	96.2	94.8	94.6	97.5

with at least one other citation (i.e. non-singletons), since these are the only citations whose segmentation we might hope to improve by using co-reference.

Table 9
Comparison of segmentation performance on non-singleton citations using entity attributes generated through co-reference vs. baseline segmentation, where co-reference is perfect

	Reinforce	Face	Reason	Constraint
Baseline	94.3	90.6	92.4	93.6
With co-reference	94.9	91.0	93.3	94.8
Error reduction	10.9	3.8	9.4	17.9

Table 10
Comparison of segmentation performance on non-singleton citations using entity attributes generated through co-reference vs. baseline segmentation, where coreference is generated automatically

	Reinforce	Face	Reason	Constraint
Baseline	94.3	90.8	92.9	93.4
With co-reference	94.9	91.4	93.5	94.3
Error reduction	10.1	6.2	9.0	14.2
p-Value	.0442	.0014	.0001	.0001

However, in practice, true co-reference information is not available. We have to resort to clustering algorithms to automatically generate co-reference information. We use a graph partition algorithm for co-reference resolution (McCallum & Wellner, 2003) which gives very good results on our datasets (above 94% F1-measure) (Wellner, McCallum, Peng, & Hay, 2004). We apply our co-reference IE system based on this co-reference results.

Table 10 shows the improved segmentation performance using machine generated co-reference. We obtain 6.2–14.2% improvement on the four datasets To test the significance of the improvements, we use McNemar's test on labeling disagreements (Gillick & Cox, 1989). At the 95% confidence level (p-value smaller than 0.05), the improvements on the four datasets are statistically significant.

Interestingly, machine generated co-reference achieves comparable improvements to true co-reference, which is supposed to be an upper bound. This can be attributed to two reasons. First, since we only improve citation segmentations with multiple citations and ignore the singletons (in which case no co-referential information is available), the numbers reported in Tables 9 and 10 are not on exactly the same data. The true upper bound for machine generated co-reference could be higher. Second, this also indicates that our co-reference performance is good enough for the purpose of co-reference information extraction and is useful in real situations.

6.6. Analysis

6.6.1. Overall performance comparison

From Tables 5–7, one can see that CRF performs significantly better than HMM, which again supports the previous findings (Lafferty et al., 2001; Pinto et al., 2003). CRFs also achieves significant better results than previously reported SVM best results, yielding new state of the art performance on this task.[3] CRFs increase the performance on nearly all the fields. The overall word accuracy is improved from 92.9% to 98.3%, which corresponds to a **78**% error rate reduction. However, as we can see word accuracy can be misleading since the HMM model even has a higher word accuracy than the SVM, although it performs much worse than SVM in most individual fields except *abstract*. Interestingly, the HMM performs much better on the *abstract* field (98% versus 93.8% F-measure) which pushes the overall accuracy up. A better comparison can be made by comparing the field-based F-measures. Here, in comparison to the SVM, CRFs improve the F1 measure from 89.7% to 93.9%, an error reduction of **36**%.

[3] It should be noted that the results on SVM were obtained from (Han et al., 2003). We did not conduct experiments on SVM using exact the same features. The comparison made here may not be exactly fair.

Table 11
Regularization comparisons: *Gaussian infinity* is non-regularized, *Gaussian variance = X* is setting variance to be *X*, *Gaussian cut 7* refers to the threshold cut method, *Gaussian divide count* refers to the divide count, *Gaussian bin N* refers to the bin based method with bin size equals *N*, as described in Section 3.1

Gaussian infinity	93.3
Gaussian variance = 0.1	91.8
Gaussian variance = 0.5	93.0
Gaussian variance = 5	93.7
Gaussian variance = 10	93.5
Gaussian cut 7	93.4
Gaussian divide count	92.8
Gaussian bin 5	93.6
Gaussian bin 10	92.9
Gaussian bin 15	**93.9**
Gaussian bin 20	93.2
Hyperbolic	92.8
Exponential	93.6

6.6.2. Effects of regularization

The results of different regularization methods are summarized in Table 11. Setting Gaussian variance of features depending on feature count performs better (from 93.3% to 93.9%, an error reduction of 9%). Results are averaged over five random runs. In our experiments we found the Gaussian prior to consistently perform better than the others. An exponential prior performs comparatively to the Gaussian prior. The performance of the exponential prior critically depends on the choice of α (0.1 in our case). Though the exponential prior does not give better accuracy in our case, it induces sparse solutions which require less main memory to store the active parameters. In our experiments, it reduces parameters by 2/3. The hyperbolic prior is not as effective as other alternatives.

6.6.3. Effects of exploring feature space

1. *State transition features*: We summarize the comparison of different state transition models in Table 12. The first column is the four different state transition models, the second column is the overall word accuracy of these models. Comparing the rows, one can see that the second-order model performs the best, but not significantly better than the first-order + transitions and the third-order model. However, the first-order model performs significantly worse. This is because the first-order model ignores $f(y_{t-1}, y_t)$, the state transition feature. The third order should perform better if enough training data was available.
2. *Effects of layout features*: The results of using different features are shown in Table 13. The layout feature dramatically increases the performance, raising the F1 measure from 88.8% to 93.4%, whole sentence accuracy from 40.1% to 72.4%. Adding lexicon features alone improves the performance. However, when combining lexicon features and layout features, the performance is worse than that with layout features alone. The reason is that a line break is a good sign of field change. When using lexicon features, some words that occur at the beginning of lines are pushed to be *authors*, for example, though they should be *affiliation*. Our current author lexicon contains 280,351 authors which might be too general. A more representative author lexicon might avoid hurting the performance. The improvement in performance due to layout features motivates us to design a new dataset **H2** which contains font information. We expect font change to be a good indicator of field change. The results of font features are shown in Table 14, which confirms the effect of layout features in this meta-data extraction task.

Table 12
Effects of using state transitions on **H1**

First-order	89.0
First-order + trans	95.6
Second-order	96.0
Third-order	95.3

Table 13
Results of using different features on **H1**

	Word acc.	F1	Inst. acc.
Local feature	96.5	88.8	40.1
+ lexicon	96.9	89.9	53.1
+ line break layout feature	98.2	93.4	72.4
+ line break layout + lexicon	98.0	93.0	71.7

Table 14
Results of using font features on **H2**

	Word acc.	F1	Inst. acc.
No font feature	98.6	91.1	74.7
Font feature	98.7	93.1	76.0

Table 15
Labeling confusion matrix on **H1**

	title	auth.	pubnum	date	abs.	aff.	addr.	email	deg.	note	ph.	intro	k.w.	web
title	3446	0	6	0	22	0	0	0	9	25	0	0	12	0
author	0	2653	0	0	7	13	5	0	14	41	0	0	12	0
pubnum	0	14	278	2	0	2	7	0	0	39	0	0	0	0
date	0	0	3	336	0	1	3	0	0	18	0	0	0	0
abstract	0	0	0	0	53262	0	0	1	0	0	0	0	0	0
affil.	19	13	0	0	10	3852	27	0	28	34	0	0	0	1
address	0	11	3	0	0	35	2170	1	0	21	0	0	0	0
email	0	0	1	0	12	2	3	461	0	2	2	0	15	0
degree	2	2	0	2	0	2	0	5	465	95	0	0	2	0
note	52	2	9	6	219	52	59	0	5	4520	4	3	21	3
phone	0	0	0	0	0	0	0	1	0	2	215	0	0	0
intro	0	0	0	0	0	0	0	0	0	32	0	625	0	0
keyword	57	0	0	0	18	3	15	0	0	91	0	0	975	0
web	0	0	0	0	2	0	0	0	0	31	0	0	0	294

6.6.4. Error analysis

Table 15 is the classification confusion matrix of header extraction (field *page* is not shown to save space). Most errors happen to "note" and "keyword" fields. Some errors such as the confusion between *email* and *keyword* can be easily fixed by post-processing, *email* can be recognized using regular expressions. The start of *key words* can also be mostly recognized by matching "keywords" or "key words". Errors of *note* are more difficult to fix. Increasing the amount of training data may improve the performance of distinguishing between these confusing fields.

7. Conclusions and future work

We have applied conditional random fields to information extraction from research papers, and investigated the issues of regularization and feature spaces in CRFs. We have provided an empirical exploration of a few previously-published priors for conditionally-trained log-linear models. We find that the Gaussian prior with variance depending on feature frequencies performs better than several other alternatives in the literature. Feature engineering is a key component of any machine learning solution—especially in conditionally-trained models with such freedom to choose arbitrary features—and plays an even more important role than regularization. We obtain new state-of-the-art performance in extracting standard fields from

research papers, with a significant error reduction by several metrics. We also suggest better evaluation metrics to facilitate future research in this task—especially field-F1, rather than word accuracy.

We have presented an algorithm for co-reference constraint information extraction based on segmentation uncertainty. It creates canonical citations and re-ranks citation segmentations at the same time. Experiments on four datasets show significant improvements on extraction performance.

Several issues remain to be investigated. The critical component of co-reference constraint information extraction is canonical citation creation, which is an open data integration problem. The performance of co-reference IE largely depends on the quality of canonical citations. Given the large potential room for improvement using N-best list as seen in Table 8, a better canonical creation model is worth future investigation. Our work is incorporated into a new paper search engine REXA, available at http://rexo.info.

Acknowledgements

We sincerely thank Marjorie Freedman and Linnea Micciulla for proof reading. This work was supported in part by the Center for Intelligent Information Retrieval, in part by SPAWARSYSCEN-SD grant number N66001-02-1-8903, in part by the National Science Foundation Cooperative Agreement number ATM-9732665 through a subcontract from the University Corporation for Atmospheric Research (UCAR) and in part by The Central Intelligence Agency, the National Security Agency and National Science Foundation under NSF grant # IIS-0326249. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor.

References

Basu, C., Hirsh, H., Cohen, W., & Neville-Manning, C. (2001). Technical paper recommendation: A study in combining multiple information sources. *Journal of Artificial Intelligence Research (JAIR), 14*, 231–252.

Chen, S., & Rosenfeld, R. (2000). A survey of smoothing techniques for ME models. *IEEE Trans. Speech and Audio Processing, 8*(1), 37–50.

Cohen, W. (2003). Secondstring. http://secondstring.sourceforge.net/.

Cohen, W., McAllester, D., & Kautz, H. (2000). Hardening soft information sources. In *KDD 2000* (pp. 255–259). ACM.

Collins, M. (2000). Discriminative reranking for natural language parsing. In *Proceedings of the seventeenth international conference on machine learning (IGML)* (pp. 175–182).

Collins, M. (2002). Ranking algorithms for named-entity extraction: Boosting and the voted perceptron. In *Proceedings of 40th anniversary meeting of association of computational linguistics (ACL)*.

Gillick, L., & Cox, S. (1989). Some statistical issues in the comparison of speech recognition algorithms. In *Proceedings of the international conference on acoustics speech and signal processing (ICASSP)* (pp. 532–535).

Goodman, J. (2003). Exponential priors for maximum entropy models. Technical Report, MSR Technical Report.

Han, H., Giles, C., Manavoglu, E., Zha, H., Zhang, Z., & Fox, E. (2003). Automatic document meta-data extraction using support vector machines. In *Proceedings of joint conference on digital libraries (JCDL)*.

Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning (ICML)*.

Lawrence, S., Giles, C. L., & Bollacker, K. (1999). Digital libraries and autonomous citation indexing. *IEEE Computer, 32*(6), 67–71.

Malouf, R. (2002). A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of the sixth conference on natural language learning (CoNLL)*.

McCallum, A. (2003). Efficiently inducing features of conditional random fields. In *Proceedings of 19th conference on uncertainty in artificial intelligence (UAI)*.

McCallum, A., & Li, W. (2003). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of seventh conference on natural language learning (CoNLL)*.

McCallum, A., Nigam, K., Rennie, J., & Seymore, K. (2000). Automating the construction of internet portals with machine learning. *Information Retrieval Journal, 3*, 127–163.

McCallum, A., & Wellner, B. (2003). Toward conditional models of identity uncertainty. In *Proceedings of IJCAI workshop on information integration and the web*.

Ney, H., Essen, U., & Kneser, R. (1995). On the estimation of small probabilities by leaving-one-out. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 17*(12), 1202–1212.

Pietra, S., Pietra, V., & Lafferty, J. (1995). Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 19*(4).

Pinto, D., McCallum, A., Wei, X., & Croft, W. (2003). Table extraction using conditional random fields. In *Proceedings of the 26th annual international ACM SIGIR conference on research and development in information retrieval (SIGIR'03)*.

Schwartz, R., & Chow, Y. (1990). The N-best algorithm: An efficient and exact procedure for finding the N most likely sentence hypotheses. In *Proceedings of the international conference on acoustics speech and signal processing (ICASSP)*.

Seymore, K., McCallum, A., & Rosenfeld., R. (1999). Learning hidden Markov model structure for information extraction. In *Proceedings of AAAI'99 workshop on machine learning for information extraction.*

Sha, F., & Pereira, F. (2003). Shallow parsing with conditional random fields. In *Proceedings of human language technology conference and North American chapter of the association for computational linguistics (HLT-NAACL'03)*.

Takasu, A. (2003). Bibliographic attribute extraction from erroneous references based on a statistical model. In *Proceedings of joint conference on digital libraries (JCDL)*.

Wellner, B., McCallum, A., Peng, F., & Hay, M. (2004). An integrated, conditional model of information extraction and coreference with application to citation matching. In *Proceedings of 20th conference on uncertainty in artificial intelligence (UAI 2004)*. Baff, Canada.