Generalized Expectation Criteria for Bootstrapping Extractors using Record-Text Alignment

Kedar Bellare
Dept. of Computer Science
University of Massachusetts
Amherst, MA 01003
kedarb@cs.umass.edu

Andrew McCallum
Dept. of Computer Science
University of Massachusetts
Amherst, MA 01003
mccallum@cs.umass.edu

Abstract

Traditionally, machine learning approaches for information extraction require human annotated data that can be costly and time-consuming to produce. However, in many cases, there already exists a database (DB) with schema related to the desired output, and records related to the expected input text. We present a conditional random field (CRF) that aligns tokens of a given DB record and its realization in text. The CRF model is trained using only the available DB and unlabeled text with generalized expectation criteria. An annotation of the text induced from inferred alignments is used to train an information extractor. We evaluate our method on a citation extraction task in which alignments between DBLP database records and citation texts are used to train an extractor. Experimental results demonstrate an error reduction of 35% over a previous state-of-the-art method that uses heuristic alignments.

1 Introduction

A substantial portion of information on the Web consists of unstructured and semi-structured text. Information extraction (IE) systems segment and label such text to populate a structured database that can then be queried and mined efficiently.

In this paper, we mainly deal with information extraction from text fragments that closely resemble structured records. Examples of such texts include citation strings in research papers, contact addresses on person homepages and apartment listings in classified ads. Pattern matching and rule-based approaches for IE (Brin, 1998; Agichtein and Gravano, 2000; Etzioni et al., 2005) that only use specific patterns, and delimiter and font-based cues for segmentation are prone to failure on such data because these cues are generally not broadly reliable. Statistical machine learning methods such as hidden Markov models (HMMs) (Rabiner, 1989; Seymore et al., 1999; Freitag and McCallum, 1999) and conditional random fields (CRFs) (Lafferty et al., 2001; Peng and McCallum, 2004; Sarawagi and Cohen, 2005) have become popular approaches to address the text extraction problem. However, these methods require labeled training data, such as annotated text, which is often scarce and expensive to produce.

In many cases, however, there already exists a database with schema related to the desired output, and records that are imperfectly rendered in the available unlabeled text. This database can serve as a source of significant supervised guidance to machine learning methods. Previous work on using databases to train information extractors has taken one of three simpler approaches. In the first, a separate language model is trained on each column of the database and these models are then used to segment and label a given text sequence (Agichtein and Ganti, 2004; Canisius and Sporleder, 2007). However, this approach does not model context, errors or different formats of fields in text, and requires large number of database entries to learn an accurate language model. The second approach (Sarawagi and Cohen, 2004; Michelson and Knoblock, 2005; Mansuri and Sarawagi, 2006) uses database or dictionary lookups in combination with similarity measures to add features to the text sequence. Although these features are very informative, learning algorithms still require annotated data to make use of them. The final approach heuristically labels texts using matching records and learns extractors from these annotations (Ramakrishnan and Mukherjee, 2004; Bellare and McCallum, 2007; Michelson and Knoblock, 2008). Heuris-

tic labeling decisions, however, are made independently without regard for the Markov dependencies among labels in text and are sensitive to subtle changes in text.

Here we propose a method that automatically induces a labeling of an input text sequence using a word *alignment* with a matching database record. This induced labeling is then used to train a text extractor. Our approach has several advantages over previous methods. First, we are able to model field ordering and context around fields by learning an extractor from annotations of the text itself. Second, a probabilistic model for word alignment can exploit dependencies among alignments, and is also robust to errors, formatting differences, and missing fields in text and the record.

Our word alignment model is a conditional random field (CRF) (Lafferty et al., 2001) that generates alignments between tokens of a text sequence and a matching database record. The structure of the graphical model resembles IBM Model 1 (Brown et al., 1993) in which each target (record) word is assigned one or more source (text) words. The alignment is generated conditioned on both the record and text sequence, and therefore supports large sets of rich and non-independent features of the sequence pairs. Our model is trained without the need for labeled word alignments by using generalized expectation (GE) criteria (Mann and McCallum, 2008) that penalize the divergence of specific model expectations from target expectations. Model parameters are estimated by minimizing this divergence. To limit over-fitting we include a $\mathbb{L}_2$-regularization term in the objective. The model expectations in GE criteria are taken with respect to a set of alignment latent variables that are either specific to each sequence pair (local) or summarizing the entire data set (global). This set is constructed by including all alignment variables a that satisfy a certain binary feature (e.g., $f(a, \mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2) = 1$, for labeled record $(\mathbf{x}_1, \mathbf{y}_1)$, and text sequence $\mathbf{x}_2$). One example global criterion is that "an alignment exists between two orthographically similar[1] words 95% of the time." Here the criterion has a *target expectation* of 95% and is defined over alignments $\{a = \langle i, j \rangle \mid \mathbf{x}_1[i] \sim \mathbf{x}_2[j], \forall \mathbf{x}_1, \mathbf{x}_2\}$. Another criterion for extraction can be "the word 'EMNLP' is always aligned with the record label *booktitle*".

[1]Two words are orthographically similar if they have low edit distance.

Record

name	address	city	state	phone
restaurant katsu	*n. hillhurst avenue*	*los angeles*		*665-1891*

Text

katzu, 1972 hillhurst ave., los feliz, california

Table 1: An example of a matching record-text pair for restaurant addresses.

This criterion has a *target* of 100% and defined for $\{a = \langle i, j \rangle \mid \mathbf{y}_1[i] = \textit{booktitle} \wedge \mathbf{x}_2[j] =$ 'EMNLP', $\forall \mathbf{y}_1, \mathbf{x}_2\}$. One-to-one correspondence between words in the sequence pair can be specified as collection of local expectation constraints. Since we directly encode prior knowledge of how alignments behave in our criteria, we obtain sufficiently accurate alignments with little supervision.

We apply our method to the task of citation extraction. The input to our training algorithm is a set of matching DBLP[2]-record/citation-text pairs and global GE criteria[3] of the following two types: (1) *alignment* criteria that consider features of mapping between record and text words, and, (2) *extraction* criteria that consider features of the schema label assigned to a text word. In our experiments, the parallel record-text pairs are collected manually but this process can be automated using systems that match text sequences to records in the DB (Michelson and Knoblock, 2005; Michelson and Knoblock, 2008). Such systems achieve very high accuracy close to 90% F1 on semi-structured domains similar to ours.[4] Our trained alignment model can be used to directly align new record-text pairs to create a labeling of the texts. Empirical results demonstrate a 20.6% error reduction in token labeling accuracy compared to a strong baseline method that employs a set of high-precision alignments. Furthermore, we provide a 63.8% error reduction compared to IBM Model 4 (Brown et al., 1993). Alignments learned by our model are used to train a linear-chain CRF extractor. We obtain an error reduction of 35.1% over a previous state-of-the-art extraction method that uses heuristically generated alignments.

[2]http://www.informatik.uni-trier.de/~ley/db/

[3]Expectation criteria used in our experiments are listed at *http://www.cs.umass.edu/~kedarb/dbie_expts.txt*.

[4]To obtain more accurate record-text pairs, matching methods can be tuned for high precision at the expense of recall. Alternatively, humans can cheaply provide match/mismatch labels on automatically matched pairs.

2 Record-Text Alignment

Here we provide a brief description of the record-text alignment task. For the sake of clarity and space, we describe our approach on a *fictional* restaurant address data set. The input to our system is a database (DB) consisting of records (possibly containing errors) and corresponding texts that are realizations of these DB records. An example of a matching record-text pair is shown in Table 1. This example displays the differences between the record and text: (1) spelling errors: *katsu* → *katzu*, (2) word insertions (*restaurant*), deletions (*1972*), substitutions (*angeles* → *feliz*), (3) abbreviations (*avenue* → *ave.*), (4) missing fields in text (**phone**=*665-1891*), and (5) extra fields in text (**state**=*california*). These discrepancies plus the unknown ordering of fields within text can be addressed through word alignment.

	katzu,	*1972*	*hillhurst*	*ave.,*	*los*	*feliz,*	*california*
restaurant **[name]**	.	.	.	.	.	.	.
katsu **[name]**	■	.	.	.	.	.	.
null **[name]**	.	.	.	.	.	.	.
n. **[address]**	.	.	.	.	.	.	.
hillhurst **[address]**	.	.	■	.	.	.	.
avenue **[address]**	.	.	.	■	.	.	.
null **[address]**	.	■	.	.	.	.	.
los **[city]**	.	.	.	.	■	.	.
angeles **[city]**	.	.	.	.	.	■	.
null **[city]**	.	.	.	.	.	.	.
null **[state]**	.	.	.	.	.	.	■
665-1891 **[phone]**	.	.	.	.	.	.	.
null **[phone]**	.	.	.	.	.	.	.

Table 2: Example of a word alignment. ■ represents aligned tokens. Vertical text at the bottom are the text tokens. Horizontal text on the left are tokens from the DB record with labels shown in braces.

An example word alignment between the record and text is shown in Table 2. Tokenization of record/text string is based on whitespace characters. We add a special *null* token at the field boundaries for each label in the schema to model word insertions. The record sequence is obtained by concatenating individual fields according to the DB schema order. As in statistical word alignment, we assume the DB record to be our source and the text to be our target. The induced labeling of the text is given by (**name**, **address**, **address**, **address**, **city**, **city**, **state**) which can be used to train an information extractor. In the next section, we present our approach to address this task.

3 Approach

We first define notation that will be used throughout this section. Let $(\mathbf{x}_1, \mathbf{y}_1)$ be a database record with token sequence $\mathbf{x}_1 = \langle x_1[1], x_1[2], \ldots, x_1[m] \rangle$ and label sequence $\mathbf{y}_1 = \langle y_1[1], y_1[2], \ldots, y_1[m] \rangle$ with $y_1[*] \in \mathcal{Y}$ where $\mathcal{Y}$ is the database schema. Let $\mathbf{x}_2 = \langle x_2[1], x_2[2], \ldots, x_2[n] \rangle$ be the text sequence. Let $\mathbf{a} = \langle a_1, a_2, \ldots, a_n \rangle$ be an alignment sequence of same length as the target text sequence. The alignment $a_i = j$ assigns the DB token-label pair $(x_1[j], y_1[j])$ to the text token $x_2[i]$.

3.1 Conditional Random Field for Alignment

Our conditional random field (CRF) for alignment has a graphical model structure that resembles that of IBM Model 1 (Brown et al., 1993). The CRF is an undirected graphical model that defines a probability distribution over alignment sequences $\mathbf{a}$ conditioned on the inputs $(\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2)$ as:

$$p_\Theta(\mathbf{a}|\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2; \Theta) = \frac{\exp(\sum_{t=1}^{n} \Theta^\top \vec{f}(a_t, \mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, t)}{Z_\Theta(\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2)}, \quad (1)$$

where $\vec{f}(a_t, \mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, t)$ are feature functions defined over the alignments and inputs, Θ are the model parameters and $Z_\Theta(\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2) = \sum_{\mathbf{a}'} \exp(\sum_{t=1}^{n} \Theta^\top \vec{f}(a'_t, \mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, t)$ is the partition function.

The feature vector $\vec{f}(a_t, \mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, t)$ is the concatenation of two types of feature functions: (1) *alignment* features $f_{align}(a_t, \mathbf{x}_1, \mathbf{x}_2, t)$ defined on source-target tokens, and, (2) *extraction* features $f_{extr}(a_t, \mathbf{y}_1, \mathbf{x}_2, t)$ defined on source labels and target text. To obtain the probability of an alignment in a particular position t we marginalize out the alignments over the rest of the positions $\{1, \ldots, n\} \backslash \{t\}$,

$$p_\Theta(a_t|\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2) = \sum_{\{a[1\ldots n]\} \backslash \{a_t\}} p_\Theta(\mathbf{a}|\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2)$$
$$= \frac{\exp(\Theta^\top \vec{f}(a_t, \mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, t))}{\exp(\sum_{a'} \Theta^\top \vec{f}(a', \mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, t))} \quad (2)$$

Furthermore, the marginal over label y_t assigned to the text token $x_2[t]$ at time step t during align-

ment is given by

$$p_\Theta(y_t|\mathbf{x}_2) = \sum_{\{a_t|\mathbf{y}_1[a_t]=y_t\}} p_\Theta(a_t|\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2), \quad (3)$$

where $\{a_t \mid \mathbf{y}_1[a_t] = y_t\}$ is the set of alignments that result in a labeling y_t for token $x_2[t]$. Henceforth, we abbreviate $p_\Theta(a_t|\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2)$ to $p_\Theta(a_t)$. The gradient of $p_\Theta(a_t)$ with respect to parameters Θ is given by

$$\frac{\partial p_\Theta(a_t)}{\partial \Theta} = p_\Theta(a_t) \left[\vec{f}(a_t, \mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, t) - E_{p_\Theta(a)} \left(\vec{f}(a, \mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, t) \right) \right], \quad (4)$$

where the expectation term in the above equation sums over all alignments a at position t. We use the Baum-Welch and Viterbi algorithms to compute marginal probabilities and best alignment sequences respectively.

3.2 Expectation Criteria and Parameter Estimation

Let $\mathcal{D} = \langle (\mathbf{x}_1^{(1)}, \mathbf{y}_1^{(1)}, \mathbf{x}_2^{(1)}), \ldots, (\mathbf{x}_1^{(K)}, \mathbf{y}_1^{(K)}, \mathbf{x}_2^{(K)}) \rangle$ be a data set of K record-text pairs gathered manually or automatically through matching (Michelson and Knoblock, 2005; Michelson and Knoblock, 2008). A global expectation criterion is defined on the set of alignment latent variables $\mathbf{A}_f = \{a | f(a, \mathbf{x}_1^{(i)}, \mathbf{y}_1^{(i)}, \mathbf{x}_2^{(i)}) = 1, \forall i = 1 \ldots K\}$ on the entire data set that satisfy a given binary feature $f(a, \mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2)$. Similarly a local expectation criterion is defined only for a specific instance $(\mathbf{x}_1^{(i)}, \mathbf{y}_1^{(i)}, \mathbf{x}_2^{(i)})$ with the set $\mathbf{A}_f = \{a | f(a, \mathbf{x}_1^{(i)}, \mathbf{y}_1^{(i)}, \mathbf{x}_2^{(i)}) = 1\}$. For a feature function f, a target expectation p, and, a weight w, our criterion minimizes the squared divergence

$$\Delta(f, p, w, \Theta) = w \left(\frac{E_{p_\Theta}(\mathbf{A}_f)}{|\mathbf{A}_f|} - p \right)^2, \quad (5)$$

where $E_{p_\Theta}(\mathbf{A}_f) = \sum_{a \in \mathbf{A}_f} p_\Theta(a)$ is the sum of marginal probabilities given by Equation (2) and $|\mathbf{A}_f|$ is the size of the variable set. The weight w influences the importance of satisfying a given expectation criterion. Equation (5) is an instance of generalized expectation criteria (Mann and McCallum, 2008) that penalizes the divergence of a specific model expectation from a given target value. The gradient of the divergence with respect to Θ is given by,

$$\frac{\partial \Delta(f, p, w, \Theta)}{\partial \Theta} = 2w \left(\frac{E_{p_\Theta}(\mathbf{A}_f)}{|\mathbf{A}_f|} - p \right) \times \left[\frac{1}{|\mathbf{A}_f|} \sum_{a \in \mathbf{A}_f} \frac{\partial p_\Theta(a)}{\partial \Theta} - p \right], \quad (6)$$

where the gradient $\frac{\partial p_\Theta(a)}{\partial \Theta}$ is given by Eq. (4). Given expectation criteria $\mathcal{C} = \langle \mathbf{F}, \mathbf{P}, \mathbf{W} \rangle$ with a set of binary feature functions $\mathbf{F} = \langle f_1, \ldots, f_l \rangle$, target expectations $\mathbf{P} = \langle p_1, \ldots, p_l \rangle$ and weights $\mathbf{W} = \langle w_1, \ldots, w_l \rangle$, we maximize the objective

$$\mathcal{O}(\theta; \mathcal{D}, \mathcal{C}) = \max_\Theta - \sum_{i=1}^{l} \Delta(f_i, p_i, w_i, \Theta) - \frac{||\Theta||^2}{2}, \quad (7)$$

where $||\Theta||^2/2$ is the regularization term added to limit over-fitting. Hence the gradient of the objective is

$$\frac{\partial \mathcal{O}(\theta; \mathcal{D}, \mathcal{C})}{\partial \Theta} = - \sum_{i=1}^{l} \frac{\partial \Delta(f_i, p_i, w_i, \Theta)}{\partial \Theta} - \Theta.$$

We maximize our objective (Equation 7) using the L-BFGS algorithm. It is sometimes necessary to restart maximization after resetting the Hessian calculation in L-BFGS due to non-convexity of our objective.[5] Also, non-convexity may lead to a local instead of a global maximum. Our experiments show that local maxima do not adversely affect performance since our accuracy is within 4% of a model trained with gold-standard labels.

3.3 Linear-chain CRF for Extraction

The alignment CRF (**AlignCRF**) model described in Section 3.1 is able to predict labels for a text sequence given a matching DB record. However, without corresponding records for texts the model does not perform well as an extractor because it has learned to rely on the DB record and alignment features (Sutton et al., 2006). Hence, we train a separate linear-chain CRF on the alignment-induced labels for evaluation as an extractor.

The extraction CRF (**ExtrCRF**) employs a fully-connected state machine with a unique state per label $y \in \mathcal{Y}$ in the database schema. The CRF induces a conditional probability distribution over label sequences $\mathbf{y} = \langle y_1, \ldots, y_n \rangle$ and input text sequences $\mathbf{x} = \langle x_1, \ldots, x_n \rangle$ as

$$p_\Lambda(\mathbf{y}|\mathbf{x}) = \frac{\exp\left(\sum_{t=1}^{n} \Lambda^\top \vec{g}(y_{t-1}, y_t, \mathbf{x}, t) \right)}{Z_\Lambda(\mathbf{x})}. \quad (8)$$

[5] Our L-BFGS optimization procedure checks whether the approximate Hessian computed from cached gradient vectors is positive semi-definite. The optimization is restarted if this check fails.

In comparison to our earlier zero-order **AlignCRF** model, our **ExtrCRF** is a first-order model. All the feature functions in this model $g(y_{t-1}, y_t, \mathbf{x}, t)$ are a conjunction of the label pair (y_{t-1}, y_t) and input observational features. $Z_\Lambda(\mathbf{x})$ in the equation above is the partition function. Inference in the model is performed using the Viterbi algorithm.

Given expectation criteria $\mathcal{C}$ and data set $\mathcal{D} = \langle(\mathbf{x}_1^{(1)}, \mathbf{y}_1^{(1)}, \mathbf{x}_2^{(1)}), \ldots, (\mathbf{x}_1^{(K)}, \mathbf{y}_1^{(K)}, \mathbf{x}_2^{(K)})\rangle$, we first estimate the parameters Θ of **AlignCRF** model as described in Section 3.2. Next, for all text sequences $\mathbf{x}_2^{(i)}, i = 1 \ldots K$ we compute the marginal probabilities of the labels $p_\Theta(y_t|\mathbf{x}_2^{(i)}), \forall t$ using Equation (3). To estimate parameters Λ we minimize the KL-divergence between $p_\Theta(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{n} p_\Theta(y_t|\mathbf{x})$ and $p_\Lambda(\mathbf{y}|\mathbf{x})$ for all sequences $\mathbf{x}$,

$$KL(p_\Theta \| p_\Lambda) = \sum_{\mathbf{y}} p_\Theta(\mathbf{y}|\mathbf{x}) \log(\frac{p_\Theta(\mathbf{y}|\mathbf{x})}{p_\Lambda(\mathbf{y}|\mathbf{x})}) = H(p_\Theta(\mathbf{y}|\mathbf{x})) - \sum_{t, y_{t-1}, y_t} E_{p_\Theta(y_{t-1}, y_t)}[\Lambda^\top \vec{g}(y_{t-1}, y_t, \mathbf{x}, t)] + \log(Z_\Lambda(\mathbf{x})). \quad (9)$$

The gradient of the above equation is given by

$$\frac{\partial KL}{\partial \Lambda} = \sum_{t, y_{t-1}, y_t} E_{p_\Lambda(y_{t-1}, y_t|\mathbf{x})}[\vec{g}(y_{t-1}, y_t, \mathbf{x}, t)] - E_{p_\Theta(y_{t-1}, y_t|\mathbf{x})}[\vec{g}(y_{t-1}, y_t, \mathbf{x}, t)]. \quad (10)$$

Both the expectations can be computed using the Baum-Welch algorithm. The parameters Λ are estimated for a given data set $\mathcal{D}$ and learned parameters Θ by optimizing the objective

$$O(\Lambda; \mathcal{D}, \Theta) = \min_\Lambda \sum_{i=1}^{K} KL(p_\Theta(\mathbf{y}|\mathbf{x}_2^{(i)}) \| p_\Lambda(\mathbf{y}|\mathbf{x}_2^{(i)}) + \|\Lambda\|^2/2.$$

The objective is minimized using L-BFGS. Since the objective is convex we are guaranteed to obtain a global minima.

4 Experiments

In this section, we present details about the application of our method to citation extraction task.

Data set. We collected a set of 260 random records from the DBLP bibliographic database. The schema of DBLP has the following labels {*author, editor, address, title, booktitle, pages, year, journal, volume, number, month, url, ee, cdrom, school, publisher, note, isbn, chapter, series*}. The complexity of our alignment model depends on the number of schema labels and number of tokens in the DB record. We reduced the number of schema labels by: (1) mapping the labels *address, booktitle, journal* and *school* to *venue*, (2) mapping *month* and *year* to *date*, and (3) dropping the fields *url, ee, cdrom, note, isbn* and *chapter*, since they never appeared in citation texts. We also added the other label *O* for fields in text that are not represented in the database. Therefore, our final DB schema is {*author, title, date, venue, volume, number, pages, editor, publisher, series, O*}.

For each DBLP record we searched on the web for matching citation texts using the first author's last name and words in the title. Each citation text found is manually labeled for evaluation purposes. An example of a matching DBLP record-citation text pair is shown in Table 3. Our data set contains 522 record-text pairs for 260 DBLP entries and can be found at *http://www.cs.umass.edu/∼kedarb/dbie_cite_data.sgml*.

Features and Constraints. We use a variety of rich, non-independent features in our models to optimize system performance. The input features in our models are of the following two types:

(a) Extraction features in the **AlignCRF** model $(f(a_t, \mathbf{y}_1, \mathbf{x}_2, t))$ and **ExtrCRF** model $(g(y_{t-1}, y_t, \mathbf{x}, t))$ are conjunctions of assigned labels and observational tests on text sequence at time step t. The following observational tests are used: (1) regular expressions to detect tokens containing all characters (ALLCHAR), all digits (ALLDIGITS) or both digits and characters (ALPHADIGITS), (2) number of characters or digits in the token (NUMCHAR=3, NUMDIGITS=1), (3) domain-specific patterns for *date* and *pages*, (4) token identity, suffixes, prefixes and character n-grams, (5) presence of a token in lexicons such as "last names," "publisher names," "cities," (6) lexicon features within a window of 10, (7) regular expression features within a window of 10, and (8) token identity features within a window of 3.

(b) Alignment features in the **AlignCRF** model $(f(a_t, \mathbf{x}_1, \mathbf{x}_2, t))$ that operate on the aligned source token $\mathbf{x}_1[a_t]$ and target token $\mathbf{x}_2[t]$. Again the observational tests used for alignment are: (1) exact token match tests whether the source-target tokens are string identical, (2) approximate token

DBLP record	Citation text
[Chengzhi Li]$_{author}$ [Edward W. Knightly]$_{author}$ [Coordinated Network Scheduling: A Framework for End-to-End Services.]$_{title}$ [69-]$_{pages}$ [2000]$_{date}$ [ICNP]$_{venue}$	[C. Li]$_{author}$ [and]$_{O}$ [E. Knightly.]$_{author}$ [Coordinated network scheduling: A framework for end-to-end services.]$_{title}$ [In Proceedings of IEEE ICNP]$_{venue}$ ['00,]$_{date}$ [Osaka, Japan,]$_{venue}$ [November 2000.]$_{date}$

Table 3: Example of matching record-text pair found on the web.

match produces a binary feature after binning the Jaro-Winkler edit distance (Cohen et al., 2003) between the tokens, (3) substring token match tests whether one token is a substring of the other, (4) prefix token match returns true if the prefixes match for lengths $\{1, 2, 3, 4\}$, (5) suffix token match returns true if the prefixes match for lengths $\{1, 2, 3, 4\}$, and (6) exact and approximate token matches at offsets $\{-1, -1\}$ and $\{+1, +1\}$ around the alignment.

Thus, a conditional model lets us use these arbitrary helpful features that cannot be exploited tractably in a generative model.

As is common practice (Haghighi and Klein, 2006; Mann and McCallum, 2008), we simulate user-specified expectation criteria through statistics on manually labeled citation texts. For extraction criteria, we select for each label, the top N extraction features ordered by mutual information (MI) with that label. Also, we aggregate the alignment features of record tokens whose alignment with a target text token results in a correct label assignment. The top N alignment features that have maximum MI with this correct labeling are selected as alignment criteria. We bin target expectations of these criteria into 11 bins as $[0.05, 0.1, 0.2, 0.3, \ldots, 0.9, 0.95]$.[6] In our experiments, we set $N = 10$ and use a fixed weight $w = 10.0$ for all expectation criteria (no tuning of parameters was performed). Table 4 shows a sample of GE criteria used in our experiments.[7]

Experimental Setup. Our experiments use a 3:1 split of the data for training and testing. We repeat the experiment 20 times with different random splits of the data. We train the **AlignCRF** model using the training data and the automatically created expectation criteria (Section 3.2). We evaluate our alignment model indirectly in terms of token labeling accuracy (i.e., percentage of correctly labeled tokens in test citation data) since we do not have annotated alignments. The alignment model is then used to train a **ExtrCRF** model as described in Section 3.3. Again, we use token labeling accuracy for evaluation. We also measure F1 performance as the harmonic mean of precision and recall for each label.

Label	Feature	Prior
alignment	PREFIX_MATCH4	0.95
author	LEXICON_LASTNAME	0.6
title	WINDOW_WORD=Maintenance	0.95
venue	WINDOW_WORD=Conference	0.95
date	YEAR_PATTERN	0.95
volume	NUMDIGITS=2	0.6
number	NUMDIGITS=1	0.6
pages	PAGES_PATTERN	0.95
editor	WORD_PREFIX[2]=ed	0.95
publisher	WORD=Press	0.95
series	WORD=Notes	0.95
O	WORD=and	0.7

Table 4: Sample of expectation criteria used by our model.

4.1 Alternate approaches

We compare our method against alternate approaches that either learn alignment or extraction models from training data.

Alignment approaches. We use GIZA++ (Och and Ney, 2003) to train generative directed alignment models: **HMM** and IBM **Model4** (Brown et al., 1993) from training record-text pairs. These models are currently being used in state-of-the-art machine translation systems. Alignments between matching DB records and text sequences are then used for labeling at test time.

Extraction approaches. The first alternative (**DB-CRF**) trains a linear-chain CRF for extraction on fields of the database entries only. Each field of the record is treated as a separate labeled text sequence. Given an unlabeled text sequence, it is segmented and labeled using the Viterbi algorithm. This method is an enhanced representative for (Agichtein and Ganti, 2004) in which a language model is trained for each column of the DB.

[6] Mann and McCallum (2008) note that GE criteria are robust to deviation of specified targets from actual expectations.

[7] A complete list of expectation criteria is available at *http://www.cs.umass.edu/∼kedarb/dbie_expts.txt*.

Another alternative technique constructs partially annotated text data using the matching records and a labeling function. The labeling function employs high-precision alignment rules to assign labels to text tokens using labeled record tokens. We use exact and approximate token matching rules to create a partially labeled sequence, skipping tokens that cannot be unambiguously labeled. In our experiments, we achieve a precision of 97% and a recall of 70% using these rules. Given a partially annotated citation text, we train a linear-chain CRF by maximizing the marginal likelihood of the observed labels. This marginal CRF training method (Bellare and McCallum, 2007) (**M-CRF**) was the previous state-of-the-art on this data set. Additionally, if a matching record is available for a test citation text, we can partially label tokens and use constrained Viterbi decoding with labeled positions fixed at their observed values (**M+R-CRF** approach).

Our third approach is similar to (Mann and McCallum, 2008). We create extraction expectation criteria from labeled text sequences in the training data and uses these criteria to learn a linear-chain CRF for extraction (**MM08**). The performance achieved by this approach is an upper bound on methods that: (1) use labeled training records to create extraction criteria, and, (2) only use extraction criteria without any alignment criteria.

Finally, we train a supervised linear-chain CRF (**GS-CRF**) using the labeled text sequences from the training set. This represents an upper bound on the performance that can be achieved on our task. All the extraction methods have access to the same features as the **ExtrCRF** model.

4.2 Results

Table 5 shows the results of various alignment algorithms applied to the record-text data set. Alignment methods use the matching record to perform labeling of a test citation text. The **AlignCRF** model outperforms the best generative alignment model **Model4** (IBM Model 4) with an error reduction of 63.8%. Our conjecture is that **Model4** is getting stuck in sub-optimal local maxima during EM training since our training set only contains hundreds of parallel record-text pairs. This problem may be alleviated by training on a large parallel corpus. Additionally, our alignment model is superior to **Model4** since it leverages rich non-independent features of input sequence pairs.

	HMM	**Model4**	**AlignCRF**
accuracy	78.5%	79.8%	**92.7%**
author	92.7	94.9	**99.0**
title	93.3	95.1	**97.3**
date	69.5	66.3	**81.9**
venue	73.3	73.1	**91.2**
volume	50.0	49.2	**78.5**
number	53.5	66.3	**68.0**
pages	38.2	44.1	**88.2**
editor	22.8	21.5	**78.1**
publisher	29.7	31.0	**72.6**
series	**77.4**	77.3	74.6
O	49.6	58.8	**85.7**

Table 5: Token-labeling accuracy and per-label F1 for different alignment methods. These methods all use matching DB records at test time. Bold-faced numbers indicate the best performing model. **HMM, Model4**: generative alignment models from GIZA++, **AlignCRF**: alignment model from this paper.

Table 6 shows the performance of various extraction methods. Except **M+R-CRF**, all extraction approaches, do not use any record information at test time. In comparison to the previous state-of-the-art **M-CRF**, the **ExtrCRF** method provides an error reduction of 35.1%. **ExtrCRF** also produces an error reduction of 21.7% compared to **M+R-CRF** without the use of matching records. These reductions are significant at level $p = 0.005$ using the two-tailed t-test. Training only on DB records is not helpful for extraction as we do not learn the transition structure[8] and additional context information[9] in text. This explains the low accuracy of the **DB-CRF** method. Furthermore, the **MM08** approach (Mann and McCallum, 2008) achieves low accuracy since it does not use any alignment criteria during training. Hence, alignment information is crucial for obtaining high accuracy.

Note that we do not observe a decrease in performance of **ExtrCRF** over **AlignCRF** although we are not using the test records during decoding. This is because: (1) a first-order model in **ExtrCRF** improves performance compared to a zero-order model in **AlignCRF** and (2) the use of noisy

[8] In general, the *editor* field follows the *title* field while the *author* field precedes it.

[9] The token "Vol." generally precedes the *volume* field in text. Similarly, tokens "pp" and "pages" occur before the *pages* field.

	DB-CRF	M-CRF	M+R-CRF†	MM08	ExtrCRF	GS-CRF
accuracy	70.4%	88.9%	90.8%	73.5%	**92.8%**	96.5%
author	72.4	93.7	94.1	85.4	**98.5**	99.0
title	79.4	96.7	**98.4**	83.1	94.6	98.1
date	60.1	74.5	76.2	57.8	**81.7**	93.5
venue	67.3	89.4	**91.5**	73.2	89.8	95.9
volume	20.3	69.4	74.2	27.7	**78.9**	90.5
number	30.1	72.8	**80.8**	47.8	75.1	91.4
pages	41.4	80.9	84.5	49.6	**92.1**	94.1
editor	7.1	71.1	**79.3**	75.3	73.3	93.7
publisher	62.1	67.5	**77.2**	40.2	58.5	82.2
series	65.2	74.9	**76.3**	65.9	73.8	85.8
O	54.1	7.0	8.3	57.7	**91.9**	94.5

Table 6: Token-labeling accuracy and per-label F1 for different extraction methods. Except **M+R-CRF**†, all other approaches do not use any records at test time. Bold-faced numbers indicate the best performing model. **DB-CRF**: CRF trained on DB fields. **M+R-CRF**, **M-CRF**: CRFs trained from heuristic alignments. **ExtrCRF**: Extraction model presented in this paper. **GS-CRF**: CRF trained on human annotated citation texts.

DB records in the test set for alignment often increases extraction error.

Both our models have a high F1 value for the other label *O* because we provide our algorithm with constraints for the label *O*. In contrast, since there is no realization of the *O* field in the DB records, both **M-CRF** and **M+R-CRF** methods fail to label such tokens correctly. Our alignment model trained using expectation criteria achieves a performance of 92.7% close to gold-standard training (**GS-CRF**) (96.5%). Furthermore, **ExtrCRF** obtains an accuracy of 92.8% similar to **AlignCRF** without access to DB records due to better modeling of transition structure and context.

5 Related Work

Recent research in information extraction (IE) has focused on reducing the labeling effort needed to train supervised IE systems. For instance, Grenager et al. (2005) perform unsupervised HMM learning for field segmentation, and bias the model to prefer self-transitions and transitions on boundary tokens. Unfortunately, such unsupervised IE approaches do not attain performance close to state-of-the-art supervised methods. Semi-supervised approaches that learn a model with only a few constraints specifying prior knowledge have generated much interest. Haghighi and Klein (2006) assign each label in the model certain prototypical features and train a Markov random field for sequence tagging from these labeled features. In constrast, our method uses GE criteria (Mann and McCallum, 2008) – allowing soft-labeling of features with target expectation values – to train conditional models with complex and non-independent input features. Additionally, in comparison to previous methods, an information extractor trained from our record-text alignments achieves accuracy of 93% making it useful for real-world applications. Chang et al. (2007) use beam search for decoding unlabeled text with soft and hard constraints, and train a model with top-K decoded label sequences. However, this model requires large number of labeled examples (e.g., 300 annotated citations) to bootstrap itself. Active learning is another popular approach for reducing annotation effort. Settles and Craven (2008) provide a comparison of various active learning strategies for sequence labeling tasks. We have shown, however, that in domains where a database can provide significant supervision, one can bootstrap accurate extractors with very little human effort.

Another area of research, related to the task described in our paper, is learning extractors from database records. These records are also known as field books and reference sets in literature (Canisius and Sporleder, 2007; Michelson and Knoblock, 2008). Both Agichtein and Ganti (2004) and Canisius and Sporleder (2007) train a language model for each database column. The language modeling approach is sensitive to word

re-orderings in text and other variability present in real-world text (e.g., abbreviation). We allow for word and field re-orderings through alignments and model complex transformations through feature functions. Michelson and Knoblock (2008) extract information from unstructured texts using a rule-based approach to align segments of text with fields in a DB record. Our probabilistic alignment approach is more robust and uses rich features of the alignment to obtain high performance.

Recently, Snyder and Barzilay (2007) and Liang et al. (2009) have explored record-text matching in domains with unstructured texts. Unlike a semi-structured text sequence obtained by noisily concatenating fields from a single record, an unstructured sequence may contain fields from multiple records embedded in large amounts of extraneous text. Hence, the problems of record-text matching and word alignment are significantly harder in unstructured domains. Snyder and Barzilay (2007) achieve a state-of-the-art performance of 80% F1 on matching multiple NFL database records to sentences in the news summary of a football game. Their algorithm is trained using supervised machine learning and learns alignments at the level of sentences and DB records. In contrast, this paper presents a semi-supervised learning algorithm for learning token-level alignments between records and texts. Liang et al. (2009) describe a model that simultaneously performs record-text matching and word alignment in unstructured domains. Their model is trained in an unsupervised fashion using EM. It may be possible to further improve their model performance by incorporating prior knowledge in the form of expectation criteria.

Traditionally, generative word alignment models have been trained on massive parallel corpora (Brown et al., 1993). Recently, discriminative alignment methods trained using annotated alignments on small parallel corpora have achieved superior performance. Taskar et al. (2005) train a discriminative alignment model from annotated alignments using a large-margin method. Labeled alignments are also used by Blunsom and Cohn (2006) to train a CRF word alignment model. Our method is trained using a small number of easily specified expectation criteria thus avoiding tedious and expensive human labeling of alignments. An alternate method of learning alignment models is proposed by McCallum et al. (2005) in which the training set consists of sequence pairs classified as match or mismatch. Alignments are learned to identify the class of a given sequence pair. However, this method relies on carefully selected negative examples to produce high-accuracy alignments. Our method produces good alignments as we directly encode prior knowledge about alignments.

6 Conclusion and Future Work

Information extraction is an important first step in data mining applications. Earlier approaches for learning reliable extractors have relied on manually annotated text corpora. This paper presents a novel approach for training extractors using alignments between texts and existing database records. Our approach achieves performance close to supervised training with very little supervision.

In the future, we plan to surpass supervised accuracy by applying our method to millions of parallel record-text pairs collected automatically using matching. We also want to explore the addition of Markov dependencies into our alignment model and other constraints such as monotonicity and one-to-one correspondence.

Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval 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 authors' and do not necessarily reflect those of the sponsor.

References

Eugene Agichtein and Venkatesh Ganti. 2004. Mining reference tables for automatic text segmentation. In *KDD*.

Eugene Agichtein and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *ICDL*.

Kedar Bellare and Andrew McCallum. 2007. Learning extractors from unlabeled text using relevant databases. In *IIWeb workshop at AAAI 2007*.

Phil Blunsom and Trevor Cohn. 2006. Discriminative word alignment with conditional random fields. In *ACL*.

Sergey Brin. 1998. Extracting patterns and relations from the world wide web. In *EDBT Workshop*, pages 172–183.

Peter Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19:263–311.

Sander Canisius and Caroline Sporleder. 2007. Bootstrapping information extraction from field books. In *EMNLP-CoNLL*.

M. Chang, L. Ratinov, and D. Roth. 2007. Guiding semi-supervision with constraint-driven learning. In *ACL*, pages 280–287.

William Cohen, Pradeep Ravikumar, and Stephen Fienberg. 2003. A comparison of string distance metrics for name-matching tasks. In *IJCAI*.

O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. 2005. Unsupervised named-entity extraction from the Web: An experimental study. *Artificial Intelligence*, 165.

D. Freitag and A. McCallum. 1999. Information extraction with HMM and shrinkage. In *AAAI*.

T. Grenager, D. Klein, and C. D. Manning. 2005. Unsupervised learning of field segmentation models for information extraction. In *ACL*.

Aria Haghighi and Dan Klein. 2006. Prototype-driven learning for sequence models. In *HLT-NAACL*.

John Lafferty, Andrew McCallum, and Fernando C N Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, page 282.

P. Liang, M. I. Jordan, and D. Klein. 2009. Learning semantic correspondences with less supervision. In *Association for Computational Linguistics (ACL)*.

Gideon S. Mann and Andrew McCallum. 2008. Generalized expectation criteria for semi-supervised learning of conditional random fields. In *Proceedings of ACL'08*, pages 870–878.

I. R. Mansuri and S. Sarawagi. 2006. Integrating unstructured data into relational databases. In *ICDE*.

Andrew McCallum, Kedar Bellare, and Fernando Pereira. 2005. A conditional random field for discriminatively-trained finite-state string edit distance. In *UAI*.

Matthew Michelson and Craig A. Knoblock. 2005. Semantic annotation of unstructured and ungrammatical text. In *IJCAI*, pages 1091–1098.

Matthew Michelson and Craig A. Knoblock. 2008. Creating relational data from unstructured and ungrammatical data sources. *JAIR*, 31:543–590.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29.

Fuchun Peng and A. McCallum. 2004. Accurate information extraction from research papers using conditional random fields. In *HLT-NAACL*.

Lawrence R. Rabiner. 1989. A tutorial on hidden markov models and selected applications in speech processing. *IEEE*, 17:257–286.

Sridhar Ramakrishnan and Sarit Mukherjee. 2004. Taming the unstructured: Creating structured content from partially labeled schematic text sequences. In *CoopIS/DOA/ODBASE*, volume 2, page 909.

Sunita Sarawagi and William W. Cohen. 2004. Exploiting dictionaries in named entity extraction: combining semi-markov extraction processes and data integration methods. In *KDD*, page 89.

Sunita Sarawagi and William W. Cohen. 2005. Semi-Markov conditional random fields for information extraction. In *NIPS*.

Burr Settles and Mark Craven. 2008. An analysis of active learning strategies for sequence labeling tasks. In *EMNLP*, pages 1070–1079.

K. Seymore, A. McCallum, and R. Rosenfeld. 1999. Learning hidden markov model structure for information extraction. In *Proceedings of the AAAI Workshop on ML for IE*.

Benjamin Snyder and Regina Barzilay. 2007. Database-text alignment via structured multi-label classification. In *IJCAI*.

Charles Sutton, Michael Sindelar, and Andrew McCallum. 2006. Reducing weight undertraining in structured discriminative learning. In *HLT-NAACL*.

Ben Taskar, Simon Lacoste-Julien, and Dan Klein. 2005. A discriminative matching approach to word alignment. In *HLT-EMNLP*, pages 73–80.