Interactive Information Extraction with Constrained Conditional Random Fields

Trausti Kristjansson
Aron Culotta
Paul Viola
Andrew McCallum, University of Massachusetts - Amherst

Available at: https://works.bepress.com/andrew_mccallum/40/
Interactive Information Extraction with Constrained Conditional Random Fields

Trausti Kristjansson
Microsoft Research
Redmond, Washington
traustik@microsoft.com

Aron Culotta
Dept. of Computer Science
University of Mass. Amherst
culotta@cs.umass.edu

Paul Viola
Microsoft Research
Redmond, Washington
viola@microsoft.com

Andrew McCallum
Dept. of Computer Science
University of Mass. Amherst
mccallum@cs.umass.edu

Abstract

Information Extraction methods can be used to automatically “fill-in” database forms from unstructured data such as Web documents or email. State-of-the-art methods have achieved low error rates but invariably make a number of errors. The goal of an interactive information extraction system is to assist the user in filling in database fields while giving the user confidence in the integrity of the data. The user is presented with an interactive interface that allows both the rapid verification of automatic field assignments and the correction of errors. In cases where there are multiple errors, our system takes into account user corrections, and immediately propagates these constraints such that other fields are often corrected automatically.

Linear-chain conditional random fields (CRFs) have been shown to perform well for information extraction and other language modelling tasks due to their ability to capture arbitrary, overlapping features of the input in a Markov model. We apply this framework with two extensions: a constrained Viterbi decoding which finds the optimal field assignments consistent with the fields explicitly specified or corrected by the user; and a mechanism for estimating the confidence of each extracted field, so that low-confidence extractions can be highlighted. Both of these mechanisms are incorporated in a novel user interface for form filling that is intuitive and speeds the entry of data—providing a 23% reduction in error due to automated corrections.

Introduction

A recent study showed that as part of the process of gathering and managing information, currently 70 million workers, or 59% of working adults in the U.S., complete forms on a regular basis. Filling in forms is tedious, error-prone and time-consuming. In many cases, the data that is used to populate the fields of the form is already available in computer readable form.

The goal of this work is to reduce the burden on the user to the largest extent possible, while ensuring the integrity of the data entered into the system. One typical example is the entry of contact addresses from on-line sources such as email messages or Web pages. There are more than 20 fields in a contact database, including last name, first name, address, city, state, phone, etc. As we will show, it is possible to create automatic systems which will extract over 90% of these fields correctly from a diverse set of complex sources. Given this low error rate, the first goal of a good information extraction (IE) system is to display the extracted fields so that they can be verified rapidly. The second goal is to allow for the rapid correction of incorrect fields. It is important to realize that the fields are extracted as an interdependent set. Given the name “Charles Stanley” it is likely that the first name is Charles and the last name is “Stanley.” But, the opposite is possible as well. Given the error that the two names have been switched, naive correction systems require two corrective actions. In the interactive information extraction system described below, when the user corrects the first name field to be “Stanley,” the system then automatically changes the last name field to be “Charles.” We call this capability correction propagation.

From the perspective of user interface design, there are a number of goals, including reducing cognitive load, reducing the number of user actions (clicks and keystrokes), and speeding up the data acquisition process. An important element that is often overlooked is the confidence the user has in the integrity of the data. This is crucial to the usability of the application, as users are not tolerant of surprising errors, and will discontinue the use of an automatic semi-intelligent application if it has corrupted or missclassified information. Unfortunately such factors are often hard to quantify.

An interactive form filling system is quite different from the batch processing of data, such as for warehouse data cleaning (Borkar, Deshmukh, & Sarawagi 2000). In batch processing the set of fields extracted are determined directly and are optimized for low error rates. In contrast interactive information extraction (IIE) puts additional requirements on the information extraction system. To facilitate a natural user experience, the information extraction system must display low confidence fields and make optimal use of any corrections that the user has made.

There are a number of statistical approaches for information extraction (IE) that are more or less suited to this paradigm. The most common engineering approach is to build a set of regular expressions that extract the fields in question. Regular expressions are a poor match for interac-
tive information extraction since they cannot estimate confidence, nor can they naturally incorporate user labels and corrections. Maximum Entropy Classifiers are potentially quite powerful, since they allow for the introduction of arbitrary, potentially dependent, features. Maximum entropy classifiers can also estimate the confidence in decisions. However, each field extracted using a maximum entropy model is estimated independently. For this reason the potential for correction propagation is minimal. Conditional Random Fields, a generalization both of maximum entropy models and hidden Markov models, allow for the introduction of arbitrary non-local features and capture the dependencies between labels. CRFs have been shown to perform well on information extraction tasks (McCallum & Li 2003; Pinto et al. 2003; McCallum 2003; Sha & Pereira 2003), and are well-suited for interactive information extraction since the confidence of the labels can be estimated and there is a natural scheme for optimally propagating user corrections.

There are two contributions of this paper. The first contribution is the introduction of the interactive information extraction framework. This includes a user interface that highlights the label assigned to each field in the unstructured document visually while flagging low confidence labels. The interface also allows for rapid correction using “drag and drop.” Finally, the interface supports the propagation of field corrections, so that one correction will often correct many errors.

The second contribution is a pair of new algorithms for the estimation of field confidence in CRFs and for the incorporation of constraints into the Viterbi decoding process. In this case the constraints come from corrections to incorrect fields or from the new field labels added by the user. The remainder of this paper describes each contribution in turn. We then describe a set of experiments in the domain of contact address entry. In these experiments we compare the performance of several well known algorithms against CRFs. We then investigate the effectiveness of constrained Viterbi decoding after correcting the least confident error.

User Interaction Models

The idea explored in this paper is that of populating the fields of a contact database, sometimes called a Digital Address Book. With the increase of personal digital devices such as personal digital assistants (PDAs), and cell phones, there is increasing demand for better tools for contact entry.

User Interfaces for Information Extraction

Figure 1 shows a user interface that facilitates interactive information extraction. The fields to be populated are on the left side, and the source text was pasted by the user into the right side. The information extraction system extracts text segments from the unstructured text and populates the corresponding fields in the contact record. This user interface is designed with the strengths and weaknesses of the information extraction technology in mind. Some important aspects are:

- The UI displays visual aids that allow the user to quickly verify the correctness of the extracted field. In this case color-coded correspondence is used (e.g. blue for all phone information, and yellow for email addresses). Other options include arrows or floating overlayed tags.
- The UI allows for rapid correction. For example, text segments can easily be grouped into blocks to allow for a single click-drag-drop. In the contact record at the left, field drop down menus with other candidates for the field. Alternatively the interface could include “try again” buttons next to the fields that cycle through possible alternative extractions for the field until the correct value is found.
- The UI visually alerts the user to fields that have low confidence. Furthermore, in the unstructured text box, possible alternatives may be highlighted (e.g. alternate names are indicated in orange).

Using a well-defined probabilistic model, such as CRF’s, we can correctly calculate confidence estimates for each field assignment. Estimation of confidence scores is discussed in the section “Confidence Estimation.”

Confidence scores can be utilized in a UI in a number of ways. Field assignments with relatively low confidence can be visually marked. If a field assignment has very low confidence, and is likely to be incorrect; we may choose not to fill in the field at all. The text that is most likely to be assigned to the field can then be highlighted in the textbox (e.g. in orange).

Another related case is when there are multiple text segments that are all equally likely to be classified as e.g. a name, then this could also be visually indicated (as is done in Figure 1).
User Interaction Models

For the purposes of quantitative evaluation we will simulate the behavior of a user during contact record entry, verification, and correction. This allows for a simpler experimental paradigm that can more clearly distinguish the values of the various technical components. A set of user studies will be reported elsewhere.

A large number of user interaction models are possible given the particulars of the interface and information extraction engine. Here we outline the basic models that will be evaluated in the experimental section.

**UIM1**: The simplest case. The user is presented with the results of automatic field assignment and has to correct all errors (i.e. no correction-propagation).

**UIM2**: Under this model, we assume an initial automatic field assignment, followed by a single randomly-chosen manual correction by the user. We then perform correction-propagation, and the user has to correct all remaining errors manually.

**UIM3**: This model is similar to UIM2. We assume an initial automatic field assignment. Next the user is asked to correct the least confident incorrect field. The user is visually alerted to the fi elds in order of confidence, until an error is found. We then perform correction-propagation and the user then has to correct all remaining errors manually.

**UIMm**: The user has to fill in all fi elds manually.

Performance Evaluation

The goal in designing a new application technology is that users see an immediate benefit in using the technology. Assuming that perfect accuracy is required, benefit is realized if the technology increases the time efficiency of users, or if it reduces the cognitive load, or both. Here we introduce an efficiency measure, called the Expected Number of User Actions, which will be used in addition to standard IE performance measures.

The Expected Number of User Actions: The Expected Number of User Actions (ENUA) measure is defined as the number of user actions (e.g. clicks) required to correctly enter all fields of a record. The Expected Number of User Actions will depend on the user interaction model. To express the Expected Number of User Actions we introduce the following notation: $P_i(j)$ is the probability distribution over the number of errors $j$ after $i$ manual corrections. This distribution is represented by the histogram in Figure 2.

Under UIM1, which does not involve correction propagation, the Expected Number of User Actions is:

$$\text{ENUA} = \sum_{n=0}^{\infty} n P_0(n)$$  \hspace{1cm} (1)

where $P_0(n)$ is the distribution over the number of incorrect fi elds (see Figure 2).

In models UIM2 and UIM3 the Expected Number of User Actions is

$$\text{ENUA}_1 = (1 - P_0(0)) + \sum_{n} n P_1(n).$$  \hspace{1cm} (2)

Figure 2: Histogram, where records fall into bins depending on how many fi elds in a record are in error. Solid bars are for CRF before any corrections. The shaded bars show the distribution after one random incorrect fi eld has been corrected. These can be used to estimate $P_0(n)$ and $P_1(n)$ respectively.

Constrained Conditional Random Fields

Conditional random fi elds (Lafferty, McCallum, & Pereira 2001) are undirected graphical models used to calculate the conditional probability of values on designated output nodes given values on designated input nodes. In the special case in which the designated output nodes of the graphical model are linked by edges in a linear chain, CRFs make a fi rst-order Markov independence assumption among output nodes, and thus correspond to fi nite state machines (FSMs). In this case CRFs can be roughly understood as conditionally-trained hidden Markov models, with additional flexibility to effectively take advantage of complex overlapping features.

Let $o = \langle o_1, o_2, \ldots, o_T \rangle$ be some observed input data sequence, such as a sequence of words in a document, (the values on $T$ input nodes of the graphical model). Let $\mathcal{S}$ be a set of FSM states, each of which is associated with a label, (such as a label $\text{LASTNAME}$). Let $s = \langle s_1, s_2, \ldots, s_T \rangle$ be some sequence of states, (the values on $T$ output nodes). CRFs define the conditional probability of a state sequence given an input sequence as

$$p_{\lambda}(s|o) = \frac{1}{Z_o} \exp \left( \sum_{t=1}^{T} \sum_{k} \lambda_k f_k(s_{t-1}, s_t, o, t) \right),$$  \hspace{1cm} (3)
where $Z_\alpha$ is a normalization factor over all state sequences, $f_k(s_{t-1}, s_t, o, t)$ is an arbitrary feature function over its arguments, and $\lambda_k$ is a learned weight for each feature function. The normalization factor, $Z_\alpha$, involves a sum over an exponential number of different possible state sequences, but because these nodes with unknown values are connected in a graph without cycles (a linear chain in this case), it kann be efficiently calculated via belief propagation using dynamic programming. Inference to find the most likely state sequence (very much like Viterbi algorithm in this case) is also a simple matter of dynamic programming.

Maximum a posteriori training of these models is efficiently performed by hill-climbing methods such as conjugate gradient, or its improved second-order cousin, limited-memory BFGS (Sha & Pereira 2003).

In order to facilitate the user interaction model, we need to clamp some of the hidden variables to particular values. Doing so results in the constrained Viterbi algorithm for CRFs, described below.

For HMMs, the Viterbi algorithm (Rabiner 1989) is an efficient dynamic programming solution to the problem of finding the state sequence most likely to have generated the observation sequence. Because CRFs are conditionally trained, the CRF Viterbi algorithm instead finds the most likely state sequence given an observation sequence,

$$s^* = \arg \max_s p_A(s|o).$$

To avoid an exponential-time search over all possible settings of $s$, Viterbi stores the probability of the most likely path at time $t$ which accounts for the first $t$ observations and ends in state $s_t$. Following the notation of Rabiner (1989), we define this probability to be $\delta_t(s_t)$, where $\delta_0(s_t)$ is the probability of starting in each state $s_t$, and the induction step is given by:

$$\delta_{t+1}(s_t) = \max_{s'} \left[ \delta_t(s') \exp \left( \sum_k \lambda_k f_k(s', s_t, o, t) \right) \right].$$

The recursion terminates in

$$p^* = \arg \max_i [\delta_T(s_i)].$$

We can backtrack through the dynamic programming table to recover $s^*$.

Constrained Viterbi alters Eq. 4 such that $s^*$ is constrained to pass through some subpath $C = \langle i, i+1, \ldots \rangle$. These constraints $C$ now define the new induction is $\delta_{t+1}(s_t) = \left\{ \begin{array}{ll}
\max_{s'} \left[ \delta_t(s') \exp \left( \sum_k \lambda_k f_k(s', s_t, o, t) \right) \right] & \text{if } s_t = s_{t+1} \\
0 & \text{otherwise} \end{array} \right.$

for all $s_{t+1} \in C$. For time steps not constrained by $C$, Eq. 4 is used instead.

In the context of interactive form filling, the constraints $C$ correspond to a set of observations (an address fi eld) manually corrected by the user. Upon correction, the system runs Constrained Viterbi to fi nd the best path that conforms to the corrected fi eld. In addition to correcting the fi eld the user indicates, this process may also change the predicted states for observations outside of the corrected fi eld. This is because the recursive formulation in Eq. 5 can affect optimal paths before and after the time steps specified in $C$.

Confidence Estimation

To estimate the confidence the CRF has in an extracted fi eld, we employ a technique we term Constrained Forward-Backward (Culotta & McCallum 2004). The Forward-Backward algorithm is similar to the Viterbi algorithm: instead of choosing the maximum state sequence, Forward-Backward evaluates all possible state sequences given the observation sequence.

The “forward values” $\alpha_{t+1}(s_i)$ are recursively defined similarly to Eq. 4, except the max is replaced by a summation. Thus we have

$$\alpha_{t+1}(s_t) = \sum_{s'} \alpha_t(s') \exp \left( \sum_k \lambda_k f_k(s', s_t, o, t) \right).$$

Furthermore, the recursion terminates to define $Z_\alpha$ in Eq. 3:

$$Z_\alpha = \sum_i \alpha_T(s_i).$$

The Constrained Forward-Backward algorithm calculates the probability of any sequence passing through a set of constraints

$$C = \langle s_1, \ldots, s_r \rangle,$

where now $s_q \in C$ can be either a positive constraint or a negative constraint. A negative constraint constrains the forward value calculation not to pass through state $s_q$.

The calculations of the forward values can be made to conform to $C$ in a manner similar to the Constrained Viterbi algorithm. If $\alpha_{t+1}^C(s_t)$ is the constrained forward value, then

$$Z_\alpha^C = \sum_i \alpha_T^C(s_i)$$

is the value of the constrained lattice. Our confidence estimate is equal to the normalized value of the constrained lattice:

$$Z_\alpha^C / Z_\alpha.$$

In the context of interactive form filling, the constraints $C$ correspond to an automatically extracted fi eld. The positive constraints specify the observation tokens labelled inside the fi eld, and the negative constraints specify the boundary of the fi eld. For example, if we use states names B-TITLE and I-JOBTITLE to label tokens that begin and continue a JOBTITLE fi eld, and the system labels observation sequence $\langle \alpha_2, \ldots, \alpha_6 \rangle$ as a JOBTITLE fi eld, then $C = \langle B-TITLE, s_3 = \ldots = s_5 = I-JOBTITLE, s_6 \neq I-JOBTITLE \rangle$.

Experiments

For training and testing we collected 2187 contact records (27,560 words) from web pages and emails and hand-labeled 25 classes of data fields.1 Some data came from pages containing lists of addresses, and about half came from disparate
A table shows the expected number of user actions (ENUA) for different error correction methods: CRF, MAXENT, CCFR, and manual. The table includes columns for Token Acc., F1, Prec, Rec, ENUA, and Change. CRF has an ENUA of 0.73, while MAXENT has 0.8843. CCRF has an ENUA of 0.63, and manual has 6.31. The change in ENUA compared to the baseline is -13.9% for CCRF and +770.0% for manual.

The user interaction model UIM3 uses confidence estimates to direct the user to correct errors. The ENUA metric does not account for time. The table shows that CCFR reduces the expected number of user actions by 13.9%.

User Interaction Evaluation:
The standard information retrieval metrics do not adequately capture the performance of an Interactive Information Extraction system. This paper has proposed a metric called the Expected Number of User Actions. Table 2 shows the Results for Constrained Conditional Random Field (CCRF), which uses confidence scores to direct the user to correct errors. The ENUA metric does not account for time.

Correction Propagation:
To examine the effectiveness of correction propagation, Table 3 shows the token accuracy of CCRF on contact records containing errors in at least two fields. One field is corrected by the user, with the hope that correction propagation will be effective. The comparison between CCRF and manual entry requires over 10 times more user actions.

Confidence estimation is used in UIM3. Recall that in this user interaction model, the system assigns confidence scores to the fi eld and asks the user to correct the least confident field. Interestingly, correcting a random field (ENUA = 0.63) seems to be slightly more informative for correction-propagation than correcting the least confident erroneous field (ENUA = 0.64). While this may seem surprising, recall that a field will have low confidence if the posterior probability of the competing classes is close to the score for the chosen class. Hence, it only requires a small amount of extra information to boost the posterior for one of the other classes and “flip” the classification. We can imagine a contrived example where there are two adjacent incorrect fields. In this case, we should correct the more confident field of the two to maximize correction propagation. This is because the field with lower confidence requires a smaller amount of extra information to correct its classification.

Under UIM3, the user may be required to verify a number of correct fields before an incorrect field is found, since the model may have low confidence in correct fields.

Another way of assessing the effectiveness of confidence is to ask how effective it is at directing the user to an incorrect token. In our experiments with CCRFs, the number of records that contained one or more errors was 276. The least confident field was truly incorrect in 226 out of those 276 records. Hence, confidence estimation correctly predicts an erroneous field 81.9% of the time. If we instead choose a token at random, then we will choose an incorrect token in 80 out of the 276 records, or 29.0%. In practice, the user only knows where the errors are, so confidence estimates can be used effectively to direct the user to an incorrect field.

The ENUA metric does not take into account the time it takes the user to scan the record and find incorrect fields. It is difficult to assess this without extensive user studies, where different strategies and visual cues are compared.

Table 1: Token accuracy and field performance for the Conditional Random Field based field extractor, and the Maximum Entropy based field extractor.

<table>
<thead>
<tr>
<th>Method</th>
<th>Token Acc.</th>
<th>F1</th>
<th>Prec</th>
<th>Rec</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRF</td>
<td>89.73</td>
<td>87.23</td>
<td>88.24</td>
<td>86.24</td>
</tr>
<tr>
<td>MAXENT</td>
<td>88.43</td>
<td>84.84</td>
<td>85.09</td>
<td>84.95</td>
</tr>
</tbody>
</table>

Table 2: The Expected Number of User Actions (ENUA) to completely enter a contact record. Notice that Constrained CRF with a random corrected field reduces the Expected Number of User Actions by 13.9%.

<table>
<thead>
<tr>
<th>Method</th>
<th>ENUA</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRF – (UIM1)</td>
<td>0.73</td>
<td>baseline</td>
</tr>
<tr>
<td>CCRF – (UIM2)</td>
<td>0.63</td>
<td>-13.9%</td>
</tr>
<tr>
<td>CCRF – (UIM3)</td>
<td>0.64</td>
<td>-11.3%</td>
</tr>
<tr>
<td>MAXENT – (UIM1)</td>
<td>0.94</td>
<td>+29.0%</td>
</tr>
<tr>
<td>Manual – (UIMm)</td>
<td>6.31</td>
<td>+770.0%</td>
</tr>
</tbody>
</table>

Table 3: The token accuracy of CCRF on contact records containing errors in at least two fields. One field is corrected by the user, with the hope that correction propagation will be effective.
automatically fix errors in other fields. Here, total accuracy is the token accuracy for the entire contact record, and uncorrected accuracy is the token accuracy of tokens not corrected by the user. Note that these accuracies are naturally lower than in Table 2 because we are only examining records with multiple errors.

These results show that having the user correct one field results in a 23% reduction in error in the remaining fields. This additional error reduction is a boon to users since they do not have to perform these corrections manually.

Confidence Estimation

In the preceding discussion, the goal of IIE has been to correctly fill in all fields of each record. A different scenario arises if we wish to reduce the labelling error rate of a large amount of data but we do not need the labelling to be error free. If we have limited man-power, we would like to maximize the efficiency or information gain from the human labeller.

This user interaction model assumes that we allow the human labeller to verify or correct a single field in each record, before going on to the next record.

As before the constrained conditional random field model is used, where Constrained Forward-Backward is used to predict the least confident extracted field. If this field is incorrect, then CCRF is supplied with the correct labelling, and correction propagation is performed using Constrained Viterbi. If this field is correct, then no changes are made, and we go on to the next record.

The experiments compare the effectiveness of verifying or correcting the least confident field i.e. CCRF - (L. CONF), to verifying or correcting an arbitrary field i.e. CCRF - (RANDOM).

Finally, CMAXENT is a Maximum Entropy classifier that estimates the confidence of each field by averaging the posterior probabilities of the labels assigned to each token in the field. As in CCRF, the least confident field is corrected if necessary.

Table 4 show results after a single field has been verified or corrected. Notice that if a random field is chosen to be verified or corrected, then the token accuracy goes to 91.7%, which is only a 19.2% reduction in error rate. If however, we verify or correct only the least confident field, the error rate is reduced by 56.18%.

This difference illustrates that reliable confidence prediction can increase the effectiveness of a human labeller. Also note that the 56% error reduction CCRF achieves over CRF is substantially greater than the 27% error reduction between CMAXENT and MAXENT.

<table>
<thead>
<tr>
<th></th>
<th>Error Reduction</th>
<th>F1</th>
<th>Prec</th>
<th>Rec</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCRF - (L. CONF.)</td>
<td>56.2%</td>
<td>94.45</td>
<td>94.84</td>
<td>94.06</td>
</tr>
<tr>
<td>CCRF - (RANDOM)</td>
<td>19.2%</td>
<td>89.72</td>
<td>90.72</td>
<td>88.75</td>
</tr>
<tr>
<td>CMAXENT</td>
<td>27.2%</td>
<td>89.4</td>
<td>90.34</td>
<td>88.48</td>
</tr>
</tbody>
</table>

Table 5: Token accuracy and field performance for interactive field labeling. CCRF - (L. CONF.) obtains a 56% reduction in error over CRF, and a 46% reduction in error over CCRF - (RANDOM).

To explicitly measure the effectiveness of the Constrained Forward-Backward algorithm for confidence estimation, Table 5 displays two evaluation measures: Pearson’s r and average precision. Pearson’s r is a correlation coefficient ranging from -1 to 1 which measures the correlation between a confidence score of a field and whether or not it is correct.

Given a list of extracted fields ordered by their confidence scores, average precision measures the quality of this ordering. We calculate the precision at each point in the ranked list where a correct field is found and then average these values. WorstCase is the average precision obtained by ranking all incorrect fields above all correct fields. Both Pearson’s r and average precision results demonstrate the effectiveness of Constrained Forward-Backward for estimating the confidence of extracted fields.

Related Work

This paper is the first of which we are aware that uses interactive information extraction with constraint propagation and confidence prediction to reduce human effort in form-filling. Several other efforts have studied efficient ways to interactively train an extraction system, which would later run without human interaction, for example (Cardie & Pierce 1998; Caruana, Hodor, & Rosenberg 2000).

Methods to visually link related components in a user interface have been explored, for example (Becker & Cleveland 1987; Swayne, Cook, & Buja 1991). The XGobi system uses color coding and “brushing” to indicate associations in various types of high dimensional data.

Many common word processors use visual cues to direct the users attention to possible errors in spelling and grammar. In (Miller & Myers 2001) the authors use a similar strategy, based on outlier detection.

Others have implemented systems for information extraction from free-text address blocks, however none using an interactive method. Borkar, Deshmukh, & Sarawagi (2000) obtains high accuracy using a HMM on a simpler and more limited set of fields (HouseNum, PO Box, Road, City, State, ZIP), which usually appear in very regular form. Similarly,
Bouckaert (2002) extracts the components of author affiliations from articles of a pharmaceutical journal.

Confidence prediction itself is also an under-studied aspect of information extraction—although it has been investigated in document classification (Bennett 2000), speech recognition (Gunawardana, Hon, & Jiang 1998), and machine translation (Gandrabur & Foster 2003). Much of the previous work in confidence estimation for information extraction comes from the active learning literature. For example, Scheffer, Decomain, & Wrobel (2001) derive confidence estimates using hidden Markov models in an information extraction system, however, they do not estimate the confidence of entire field labels, only singleton tokens. The token confidence is estimated by the difference between the probabilities of its first and second most likely labels, whereas our Constrained Forward-Backward (Culotta & McCallum 2004) considers multi-token field labels, and the full distribution of all suboptimal paths. Scheffer, Decomain, & Wrobel also explore an idea similar to Constrained Forward-Backward to perform Baum-Welch training with partially labelled data, wherein a limited number of labels provide constraints. However, these constraints are again for singleton tokens only. Constrained Viterbi has been used previously in bioinformatics to find sub-optimal alignments of RNA sequences (Zuker 1991).

**Conclusion and Future Work**

We have introduced a new system for assisting users when entering database records from unstructured data. We exploit CRFs to pre-populate the database field labels and allow natural user interaction where the system takes into account any corrections by the user. This is done by correction-propagation using the Constrained Viterbi algorithm in CRFs. Note that correction-propagation can be applied to any relational model. By calculating confidence estimates and highlighting low confidence field labels, we help the user spot any incorrect field labels. We have shown that both of these methods are quite useful. Using the system, the Expected Number of User Actions per record has been dramatically reduced, from 6.31 for manual entry to 0.63 or more than 10-fold.

**Acknowledgments**

We would like to thank David Parkinson for providing labelling and parsing tools and valuable discussions. This work was supported in part by the Center for Intelligent Information Retrieval, the Central Intelligence Agency, the National Security Agency, the National Science Foundation under NSF grant #IIS-0326249, SPAWARSYSCEN-SD grant number N66001-02-1-890, and by the Defense Advanced Research Projects Agency, through the Department of the Interior, NBC, Acquisition Services Division, under contract #NBCHD030010.

**References**


