Confidence Estimation for Information Extraction

Aron Culotta
Department of Computer Science
University of Massachusetts
Amherst, MA 01002
culotta@cs.umass.edu

Andrew McCallum
Department of Computer Science
University of Massachusetts
Amherst, MA 01002
mccallum@cs.umass.edu

Abstract

Information extraction techniques automatically create structured databases from unstructured data sources, such as the Web or newswire documents. Despite the successes of these systems, accuracy will always be imperfect. For many reasons, it is highly desirable to accurately estimate the confidence the system has in the correctness of each extracted field. The information extraction system we evaluate is based on a linear-chain conditional random field (CRF), a probabilistic model which has performed well on information extraction tasks because of its ability to capture arbitrary, overlapping features of the input in a Markov model. We implement several techniques to estimate the confidence of both extracted fields and entire multi-field records, obtaining an average precision of 98% for retrieving correct fields and 87% for multi-field records.

1 Introduction

Information extraction usually consists of tagging a sequence of words (e.g. a Web document) with semantic labels (e.g. PERSONNAME, PHONENUMBER) and depositing these extracted fields into a database. Because automated information extraction will never be perfectly accurate, it is helpful to have an effective measure of the confidence that the proposed database entries are correct. There are at least three important applications of accurate confidence estimation. First, accuracy-coverage trade-offs are a common way to improve data integrity in databases. Efficiently making these trade-offs requires an accurate prediction of correctness.

Second, confidence estimates are essential for *interactive information extraction*, in which users may correct incorrectly extracted fields. These corrections are then automatically propagated in order to correct other mistakes in the same record. Directing the user to the least confident field allows the system to improve its performance with a minimal amount of user effort. Kristjannson et al. (2004) show that using accurate confidence estimation reduces error rates by 46%.

A third use of confidence estimation is to improve performance of data mining algorithms that depend upon databases created by information extraction systems (McCallum and Jensen, 2003). Confidence estimates provide data mining applications with a richer set of "bottom-up" hypotheses, resulting in more accurate decisions. An example of this occurs in the task of *citation co-reference resolution*. An information extraction system is built to label each field of a paper citation (e.g. AUTHOR, TITLE), and then *co-reference resolution* is performed to merge disparate references to the same paper. Normally, errors in labeling the citation fields will carry over to errors in co-reference resolution. However, attaching a confidence value to each field allows the system to examine alternate labelings for less confident fields and give more importance to matching fields that have high confidence.

Sound probabilistic extraction models are most conducive to accurate confidence estimation because of their intelligent handling of uncertainty information. In this work we use conditional random fields (Lafferty et al., 2001), a type of undirected graphical model, to automatically label fields of contact records. Here, a record is an entire block of a person's contact information, and a field is one element of that record (e.g. COMPANYNAME). We implement several techniques to estimate both field confidence and record confidence, obtaining an average precision of 98% for fields and 87% for records.

2 Conditional Random Fields

Conditional random fields (Lafferty et al., 2001) are undirected graphical models to calculate the conditional probability of values on designated output nodes given val-

ues 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 first-order Markov independence assumption among output nodes, and thus correspond to finite 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 $\mathbf{o} = \langle o_1, o_2, ...o_T \rangle$ 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 $\mathscr{S}$ be a set of FSM states, each of which is associated with a label (such as COMPANYNAME). Let $\mathbf{s} = \langle s_1, s_2, ...s_T \rangle$ 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

$$p_\Lambda(\mathbf{s}|\mathbf{o}) = \frac{1}{Z_\mathbf{o}} \exp\left(\sum_{t=1}^{T}\sum_{k}\lambda_k f_k(s_{t-1}, s_t, \mathbf{o}, t)\right), \quad (1)$$

where $Z_\mathbf{o}$ is a normalization factor over all state sequences, $f_k(s_{t-1}, s_t, \mathbf{o}, t)$ is an arbitrary feature function over its arguments, and λ_k is a learned weight for each feature function. Even though the normalization factor, $Z_\mathbf{o}$, nominally involves a sum over an exponential number of different possible state sequences, because these nodes with unknown values are connected in a graph without cycles (a linear chain in this case), it can be efficiently calculated via belief propagation using dynamic programming. Inference (very much like the Viterbi algorithm in this case) is also a matter of dynamic programming.

Maximum aposteriori 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 and Pereira, 2003).

3 Field Confidence Estimation

We wish to estimate the probability that an extracted field is labeled correctly. The well-known Viterbi algorithm finds the most likely state sequence matching the observed word sequence. The word that Viterbi matches with a particular FSM state is extracted as belonging to the database field corresponding to that state. We can trivially obtain a numeric score for an entire sequence, and then turn this into a probability for the entire sequence by normalizing. However, to predict the confidence of an individual field, we desire the probability of a subsequence, *marginalizing* out the state selection for all other parts of the sequence. A specialization of Forward-Backward, termed *Constrained Forward-Backward* (CFB), gives us exactly this probability.

In hidden Markov models (Rabiner, 1989), the Viterbi algorithm 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, defined as

$$s^* = \operatorname*{argmax}_{\mathbf{s}} p_\Lambda(\mathbf{s}|\mathbf{o}).$$

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

$$\delta_{t+1}(s_i) = \max_{s'}\left[\delta_t(s')\exp\left(\sum_k \lambda_k f_k(s', s_i, \mathbf{o}, t)\right)\right]. \quad (2)$$

The recursion terminates in

$$s^* = \operatorname*{argmax}_{s_1 \le s_i \le s_N}[\delta_T(s_i)]$$

The Forward-Backward algorithm can be viewed as a generalization of the Viterbi algorithm: instead of choosing the optimal 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 as in Eq. 2, except the max is replaced by a summation. Thus we have

$$\alpha_{t+1}(s_i) = \sum_{s'}\left[\alpha_t(s')\exp\left(\sum_k \lambda_k f_k(s', s_i, \mathbf{o}, t)\right)\right]. \quad (3)$$

Furthermore, the recursion terminates to define $Z_\mathbf{o} = \sum_i \alpha_T(s_i)$ from Eq. 1.

To estimate the probaility that a field is extracted correctly, we constrain the Forward-Backward algorithm such that each path conforms to some subpath of constraints $C = \langle s_q \ldots s_r \rangle$ from time step q to r. Here, $s_q \in C$ can be either a positive constraint (the sequence *must* pass through s_q) or a *negative* constraint (the sequence must *not* pass through s_q).

In the context of information extraction, C corresponds to an automatically extracted field. The positive constraints specify the observation tokens labeled inside the field, and the negative constraints specify the boundary of the field. For example, if the system labels observation sequence $\langle o_2, \ldots, o_5 \rangle$ as a JOBTITLE field, then $C = \langle s_1 \neq$ JOBTITLE, $s_2 = \ldots = s_5 =$ JOBTITLE, $s_6 \neq$ JOBTITLE$\rangle$.

The calculations of the forward values can be made to conform to C by the recursion $\alpha'_q(s_i) =$

$$\begin{cases} \sum_{s'}\left[\alpha'_{q-1}(s')\exp\left(\sum_k \lambda_k f_k(s', s_i, \mathbf{o}, t)\right)\right] & \text{if } s_i \simeq s_q \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

for all $s_q \in C$, where the operator $s_i \simeq s_q$ means s_i conforms to constraint s_q. For time steps not constrained by C, Eq. 3 is used instead.

If $\alpha'_{t+1}(s_i)$ is the constrained forward value, then $Z'_{\mathbf{o}} = \sum_i \alpha'_T(s_i)$ is the value of the *constrained lattice*, the set of all paths that conform to C. Our confidence estimate is obtained by normalizing $Z'_{\mathbf{o}}$ using $Z_{\mathbf{o}}$.

We also implement an alternative method that uses the state probability distributions for each state in the extracted field. Let $\gamma_t(s_i) = p(s_i|o_1, \ldots, o_T)$ be the probability of being in state i at time t given the observation sequence (analogous to the γ variable in Rabiner (1989)). We define the confidence measure GAMMA to be $\prod_{i=u}^{v} \gamma_i(s_i)$, where u and v are the start and end indices of the extracted field.

4 Record Confidence Estimation

We can similarly use CFB to estimate the probability that an entire record is labeled correctly. The procedure is the same as in the previous section, except that C now specifies the labels for all fields in the record.

We also implement three alternative record confidence estimates. FIELDPRODUCT calculates the confidence of each field in the record using CFB, then multiplies these values together to obtain the record confidence. FIELDMIN instead uses the minimum field confidence as the record confidence. VITERBIRATIO uses the ratio of the probabilities of the top two Viterbi paths, capturing how much more likely s^* is than its closest alternative.

5 Reranking with Maximum Entropy

We also trained two conditional maximum entropy classifiers to classify fields and records as being labeled correctly or incorrectly. The resulting posterior probability of the "correct" label is used as the confidence measure. The approach is inspired by results from (Collins, 2000) that show discriminitive classifiers can improve the ranking of parses produced by a generative sentence parser.

After experimenting with many different features, the most informative inputs for the field confidence classifier were field length, the predicted label of the field, whether or not this field has been extracted elsewhere in this record, and the CFB confidence estimate for this field. For the record confidence classifier, we incorporated the following features: record length, whether or not two fields were tagged with the same label, and the CFB confidence estimate.

6 Experiments

We collected 2187 contact records (27,560 words) from Web pages and email and hand-labeled 25 classes of data fields.[1]

	Token Acc.	F1	Prec	Rec
CRF	87.32	84.11	85.43	82.83

Table 1: Token accuracy and field extraction performance for CRF.

The features for the CRF consist of the token text, capitalization features, 24 regular expressions over the token text (e.g. CONTAINSHYPHEN), and offsets of these features within a window of size 5. We also use 19 lexicons, including "US Last Names," "US First Names," and "State Names." Feature induction is not used in these experiments. The CRF is trained on 60% of the data, and the remaining 40% is split evenly into development and testing sets. The development set is used to train the maximum entropy classifiers, and the testing set is used to measure the accuracy of the confidence estimates. The for extraction results for CRF are shown in Table 1.

To evaluate confidence estimation, we use three methods. The first is Pearson's r, a correlation coefficient ranging from -1 to 1, that measures the correlation between a confidence score and whether or not the field (or record) is correctly labeled. The second is average precision, borrowed from the Information Retrieval community. Average precision is normally used to evaluate the ordering of a ranked list of retrieved documents. Instead of ranking documents by their relevance score, here we rank fields (and records) by their confidence score. WORSTCASE is the average precision obtained by ranking all incorrect instances above all correct instances. Tables 2 and 3 show that CFB and MAXENT are statistically similar, and that both outperform competing methods.

The third measure is a precision-recall graph. Better confidence estimates push the curve to the upper-right of the graph. Figure 1 shows that CFB and MAXENT dramatically outperform GAMMA.

7 Related Work

To the best of our knowledge, this paper is the first to use confidence estimation for information extraction. Confidence estimation has been studied in document classification (Bennett et al., 2002), where classifiers are built using meta-features of the document, such as the distribution of features across classes and the mean and variance of classifier-specific weights. Confidence measures are also used in speech recognition (Gunawardana et al.,

[1]The 25 fields are: FIRSTNAME, MIDDLENAME, LASTNAME, NICKNAME, SUFFIX, TITLE, JOBTITLE, COMPANYNAME, DEPARTMENT, ADDRESSLINE, CITY1, CITY2, STATE, COUNTRY, POSTALCODE, HOMEPHONE, FAX, COMPANYPHONE, DIRECTCOMPANYPHONE, MOBILE, PAGER, VOICEMAIL, URL, EMAIL, INSTANTMESSAGE

	Pearson's r	Avg. Prec
CFB	**.573**	**.976**
MaxEnt	**.571**	**.976**
Gamma	.418	.912
Random	.012	.858
WorstCase	–	.672

Table 2: Evaluation of confidence estimates for *field confidence*. CFB and MAXENT outperform competing methods.

	Pearson's r	Avg. Prec
CFB	**.626**	**.863**
MaxEnt	**.630**	**.867**
FieldProduct	.608	.858
FieldMin	.588	.843
ViterbiRatio	.313	.842
Random	.043	.526
WorstCase	–	.304

Table 3: Evaluation of confidence estimates for *record confidence*. CFB, MAXENT again perform best.

1998) to estimate the confidence of a recognized word by considering a list of commonly confused words. Finally, confidence measures have been calculated in machine translation (Gandrabur and Foster, 2003) by using a neural network to learn the probability of a correct word translation using text features and knowledge of alternate translations.

8 Conclusion

We have shown that CFB is a mathematically and empirically sound confidence estimator for finite state information extraction systems, providing strong correlation with correctness and obtaining an average precision of 97.6% for estimating field correctness. Interestingly, discriminitive reranking by MAXENT does not seem to improve performance, despite the benefit Collins (2000) has shown discriminiave reranking to provide generative parsers. We hypothesize this is because CRFs are already *discriminative* (not *generative*) models; furthermore, this may suggest that future discriminative parsing techniques will also have the benefits of discriminative reranking built-in directly.

References

Paul N. Bennett, Susan T. Dumais, and Eric Horvitz. 2002. Probabilistic combination of text classifiers using reliability indicators: models and results. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 207–214. ACM Press.

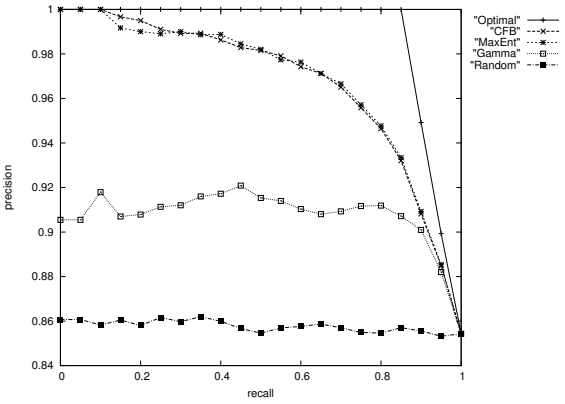

Figure 1: The precision-recall curve for *fields* shows that CFB and MAXENT outperform GAMMA.

Michael Collins. 2000. Discriminative reranking for natural language parsing. In *Proc. 17th International Conf. on Machine Learning*, pages 175–182. Morgan Kaufmann, San Francisco, CA.

Simona Gandrabur and George Foster. 2003. Confidence estimation for text prediction. In *Proceedings of the Conference on Natural Language Learning (CoNLL 2003)*, Edmonton, Canada.

A. Gunawardana, H. Hon, and L. Jiang. 1998. Word-based acoustic confidence measures for large-vocabulary speech recognition. In *Proc. ICSLP-98*, pages 791–794, Sydney, Australia.

Trausti Kristjannson, Aron Culotta, Paul Viola, and Andrew McCallum. 2004. Interactive information extraction with conditional random fields. *submitted to Nineteenth National Conference on Artificial Intelligence (AAAI 2004)*.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA.

Andrew McCallum and David Jensen. 2003. A note on the unification of information extraction and data mining using conditional-probability, relational models. In *IJCAI03 Workshop on Learning Statistical Models from Relational Data*.

L.R. Rabiner. 1989. A tutorial on hidden Markov models. In *IEEE*, volume 77, pages 257–286.

Liva Ralaivola and Florence d'Alché-Buc. 2004. Dynamical modeling with kernels for nonlinear time series prediction. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA.

Fei Sha and Fernando Pereira. 2003. Shallow parsing with conditional random fields. In Marti Hearst and Mari Ostendorf, editors, *HLT-NAACL: Main Proceedings*, pages 213–220, Edmonton, Alberta, Canada. Association for Computational Linguistics.