Conditional Estimation of HMMs for Information Extraction

Joseph Smarr
Symbolic Systems Program
Stanford University
Stanford, CA 94305-2181
jsmarr@stanford.edu

Huy Nguyen
Computer Science Dept.
Stanford University
Stanford, CA 94305-9040
htnguyen@stanford.edu

Dan Klein
Computer Science Dept.
Stanford University
Stanford, CA 94305-9040
klein@cs.stanford.edu

Christopher D. Manning
Computer Science Dept.
Stanford University
Stanford, CA 94305-9040
manning@cs.stanford.edu

Abstract

The usual procedure of optimizing hidden Markov Models for data likelihood has undesirable consequences in information extraction: it focuses attention on the data rather than on the labeling task. Often, joint likelihood is poorly correlated with extraction F_1. We demonstrate that optimizing the conditional likelihood of the target labels addresses these limitations and is more indicative of task performance. Comparing joint and conditional likelihood also helps to explain the empirical finding that, for IE, HMMs with fixed structures tend to outperform those with more flexible structures: fixed structures constrain EM to better optimize conditional likelihood.

1 Introduction

A standard task in information extraction (IE) is the *fragment extraction* task of identifying small fragments inside a larger document that pertain to a specific semantic slot of interest. For example, given a news article about a corporate acquisition, we might want to extract the name of the company that was acquired, the name of the purchasing company, and the dollar amount for which the company was acquired. Several techniques have been explored for this basic IE task, including hand-built rule-based systems (Appelt et al. 1993), wrapper induction systems (Kushmerick et al. 1997), and statistical generative models, notably hidden Markov models (HMMs) (Leek 1997, Bikel et al. 1997, Freitag and McCallum 2000).

In an HMM, the state of the hidden process encapsulates the relevant information about the past environment. In some NLP applications, such as part-of-speech tagging (Brants 2000), the states map directly onto the desired classification decisions, and the hidden process is fully observed in the training data. For example, the state over a word might encode the previous tag and the current tag, both of which are known at each point. Maximum-likelihood training is therefore trivial – parameters are estimated by taking the ratios of (smoothed) empirical counts. However, for the approach of (Freitag and McCallum 2000), which we adopt here, the states of the HMM are not fully specified in the training data. Rather, states are broken into types, such as target and background. Such models correspond to pair HMMs, the probabilistic extension of finite state transducers, which have been more explored in bioinformatics (Durbin et al. 1998: 81). In the case where the classes partition the states, this is also referred to as a class HMM (Krogh 1994). The word sequence (W) and state type sequence (C) are observed, but the states (S) themselves are not. For example, there may be 3 target states and 7 background states, the roles of which are not specified. Parameter estimation thus has the important task of deciding the roles of the states.

In the presence of incomplete data, HMMs are usually trained using the Baum-Welch algorithm (Rabiner 1989), a special case of the EM algorithm. EM is a local search procedure for optimizing the marginal likelihood of the observed data P(W,C). To the extent that EM is the only tool available, we can use it to maximize this training joint likelihood (JL), and merely hope that that it

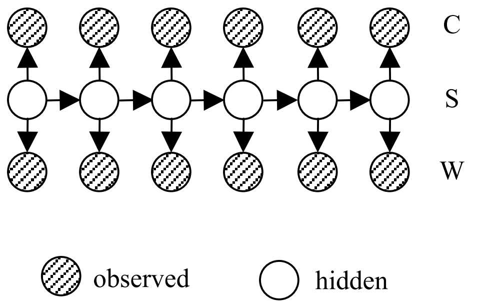

Figure 1. HMM for information extraction.

finds a felicitous configuration of the unobserved data (specific states) so that hopefully test F_1 gets optimized well enough along the way. However, for IE, we really do not care about the joint likelihood P(W,C) of the training set. Rather, for a standalone task, we typically only care about the accuracy (typically measured by F_1 of target precision and recall). If we want to use the extraction system as a stage in a probabilistic pipeline, then perhaps we care more about the conditional likelihood (CL) of the types given the words, P(C|W). Therefore, we would ideally like to maximize test F_1 or test CL. As the test set is not available, and as F_1 is a discrete measure, we settle for optimizing CL on training data.

Since CL is a continuous objective, we can examine direct optimization of this objective. The results indicate that, while training CL is better correlated with test F_1 than training JL is, direct optimization is problematic for complex problems. We illustrate these issues with simple examples, which demonstrate that the (Freitag and McCallum 2000) strategy of assigning prior semantics to states, such as prefix or suffix, is essentially human meta-optimization of CL and F_1; with these structural restrictions, direct CL optimization is easier and, more importantly, the JL optima found by EM have better CL (but worse JL) scores than in the case where structure is not constrained. Finally, we examine these issues on a real IE data set, and discuss which aspects of toy data behavior do and do not generalize.

2 An Example

In order to tractably and succinctly explore the training behavior of HMMs optimized for JL (via EM) and CL (via CG, see section 3.2) we first consider a synthetic toy data set, which is designed to reflect the relevant qualitative features of real text. The majority of each synthetic document is background text, which has regular internal structure, but which is uninformative from the perspective of identifying target fragments. Target positions are filled with distinctive words and are immediately preceded and/or followed by identifiable "prefix" and "suffix" words in the background text.

If the words in the target are completely disjoint from the words in the background, then one does not need to consider context at all to perform extraction. Two common scenarios that make information extraction tasks difficult are similar words appearing in both the target and background (e.g., company names, only some of which are companies being purchased), or several distinct targets with similar content (e.g., purchasing and purchased company, or start and end times of meetings). In such cases, a combination of distinct content and context must be identified.

Consider a simple model in which the background text consists of repeated occurrences of abc with case varying independently at random (e.g., abCaBcABc…). This is meant to be analogous to syntactic patterns in background text. We have two distinct targets types *t* and *T*, both of which show up identically in the text as X and occur relatively infrequently (roughly 2% of the document's tokens are target tokens). However, X's of type *t* are always preceded by lowercase a, whereas X's of type *T* are always preceded by uppercase A. This is meant to be analogous to the difference between a *start-time* phrase like "from 4:15" and an *end-time* phrase like "until 4:15". A sample document might look like the following (hyphen represents the class of background words):

Words: aBcabcaXcaBcAbcaBcabcAXc
Classes: -------t-------------T-

In order to correctly classify the targets, the HMM must learn that both targets emit X, but that one target is prefixed by a and the other by A. We consider a 5-state class HMM with three background states and one state for each target. That is, in the pair/class HMM there are 3 classes, two of the states are dedicated to each generating one of the two target classes, and the other three states always generate the third background class. Ini-

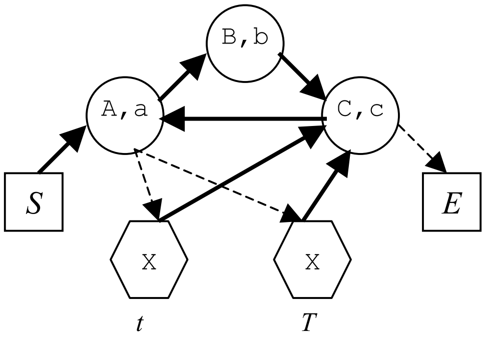

Figure 2. HMM trained jointly on toy data

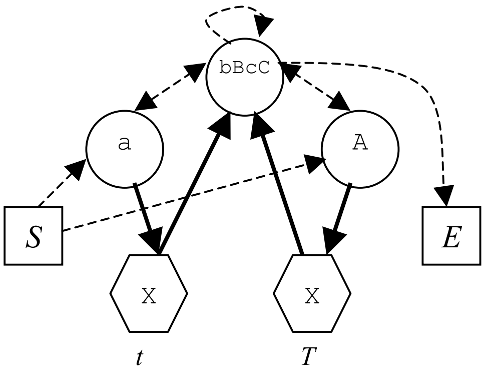

Figure 3. HMM trained conditionally on toy data

tially the transition matrix is uniform and ergodic, and all states can emit all tokens. We first train the model to maximize joint data likelihood (see section 3.1), producing the HMM shown in Figure 2. Heavy arrows indicate high-weight transitions and dashed arrows indicate low-weight transitions. Circular states are background, hexagons are targets, and there is an obligatory start and end state to mark the document boundary. Emissions are shown inside each of the states.

This HMM achieves a good joint likelihood value by using its three background states to encode the regular three-token background pattern. Notice that it has picked up on the case variation: each state emits both the uppercase and lowercase version of the letter it has specialized in. This shows EM training of an HMM effectively doing *clustering* of observations, at least in a simple case such as this. The state that generates A and a (and only this state) links to both target states, since the target prefix is always one of these tokens.

This is a very good model for explaining the word sequence—it captures the background regularity and correctly moves from the prefix states to the target states to generate the X's. But it is a useless model for the discriminative needs of the information extraction task, because nothing in the model distinguishes instances of *t* from instances of *T*. Whichever target happened to occur more frequently overall (*T* in our generated data) will end up with a slightly higher-probability transition from the Aa state, and so every X in the document will be labeled as type *T*. Since the prefix pattern is much like normal background text, it is better for the model to treat them as such than to "waste" states modeling that A transitions to *T* and a transitions to *t*.

In contrast, when the same 5-state HMM is trained to maximize conditional likelihood (see section 3.2), we end up with the HMM shown in Figure 3.

This model is qualitatively very different from the joint-maximized model. There is little if any model of the regular background pattern, because it does not increase conditional likelihood (i.e. aid in target prediction). One state generates only a and transitions strongly to *t*. The other state generates only A and transitions strongly to *T*. The third state generates B's and C's of both cases and transitions to itself as well as to the two other background states. This model has poor joint likelihood compared to the jointly trained model because it devotes none of its parameters to capturing the basic abc background pattern.[1] However, it achieves perfect F_1 on the data because it never goes to a target state from the wrong prefix. The difference between these two models is summarized in Table 1 (log likelihood (LL) and conditional log likelihood (CLL) are on training data, F_1 is on test data).

	LL	CLL	F_1
Joint HMM	-19067	-532	0.5
Cond. HMM	-27655	0	1.0

Table 1. Trained HMMs on toy data

[1] As is common with conditionally optimized models, its joint interpretation is not necessarily well-formed. By the joint likelihood of this model, we mean the unique best joint likelihood of all models with this conditionally-learned transition structure. That value is easily determined in this case because for the transition structure in figure 3, the HMM becomes a fully observed process, and its ML estimates are simply relative frequency estimates.

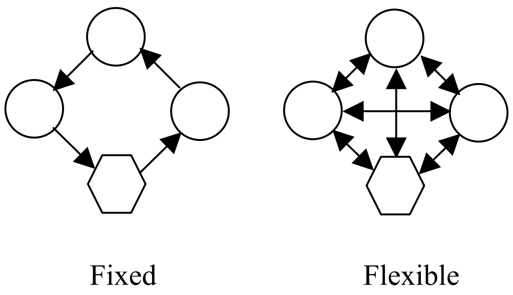

Figure 4. Simple ergodic and fixed HMM structure.

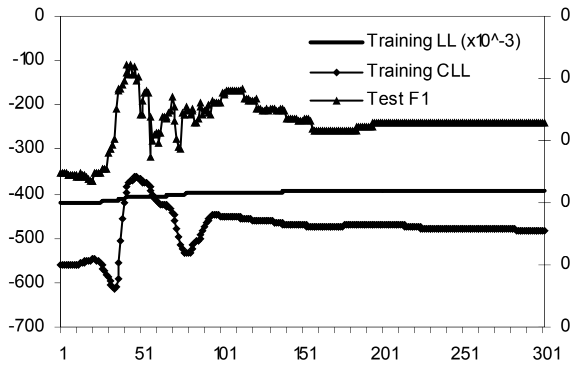

Figure 5. CLL reflects F_1 more closely than LL.

3 Scaling up to real world data

We now present HMM IE experiments performed on the *Acquisitions* data set, a collection of 600 Reuters newswire articles on the topic of corporate acquisitions, drawn from the well-known Reuters text categorization collection (Lewis 1992), and annotated with semantic tags for information extraction by Dayne Freitag (Freitag 1998). Target fields include *purchaser*, *seller*, *acquired*, and *dlramt*. Dlramt is the quantity for which the company was acquired, which usually looks like “100 mln dlrs” or “ten billion yen,” but also sometimes look like “undisclosed amount.” Dlramt is the easiest field to extract because of its distinctive content (but note that there are many other mentions of dollar amounts in the background text).

We consider two similar minimal HMM structures—an ergodic structure with three background states and one target state, and a fixed prefix/suffix structure with a background, prefix, target, and suffix state arranged in a diamond (both models are shown in Figure 4, all states also have self-transitions which are not shown). The fixed structure is a subclass of the ergodic structure that represents our intuition about a good subspace of the full ergodic parameter space. Specifically, in that subspace some transition probabilities are fixed to be 0. In practice, they are merely initialized to 0, as EM will never re-estimate a parameter away from 0.

3.1 Optimizing joint likelihood (EM)

For a general pair HMM, we cannot directly apply Baum-Welch for training. The probability distribution *P(C,S,W)* decomposes as:

$$P(\vec{C},\vec{S},\vec{W}) = \prod_i P(c_i \mid s_i)P(s_i)P(w_i \mid s_i)$$

Where the product is taken over positions along the sequence. Since *C* and *W* are observed, we will need to repeatedly find various expectations according to the posterior distribution *P(S|C,W).* When computing these expectations, we will need to sum over all state sequences *S* (this is done implicitly via dynamic programming):

$$\sum_s \prod_i P(c_i \mid s_i)P(s_i)P(w_i \mid s_i)$$

However, since $P(c_i|s_i)$ is deterministic (1 or 0), all state sequences that are inconsistent with the observed class label sequence (i.e. where at least one of the $P(c_i|s_i)$ terms is 0) will end up with 0 probability and contribute nothing to the sum. Thus we can remove the $P(c_i|s_i)$ term from the equation and instead see the observations *C* as forcing us to sum over only those state sequences that are consistent with the class labels:

$$P(\vec{C},\vec{S},\vec{W}) = \sum_{s\sim c} \prod_i P(s_i)P(w_i \mid s_i)$$

Thus the standard forward-backward algorithm need only be modified to just sum quantities over only sequences which respect the class constraints. Therefore, we can use Baum-Welch estimation with little modification.

The results of training the simple flexible- and fixed-structure HMMs are summarized in Table 2. Note that to achieve state of the art performance, one would use HMMs with more states. In this paper, however, we restrict attention to simple models whose parameters and behavior can more easily and precisely be interpreted.

	LL	CLL	F_1
Flexible	-426739	-675	0.24
Fixed	-449743	-387	0.49

Table 3. HMMs trained with EM

The fixed model converges to a worse joint likelihood than the flexible model, but its conditional likelihood is better, as is its F_1 on test data. To the extent that EM was only a device to indirectly maximize test F_1, the fixed structure seems the clear choice.

Figure 5 shows training LL and CLL along with test F_1 for the flexible model after each iteration of EM training. Two important observations are that CLL is more correlated with F_1 than LL and that the model with highest LL is not the most desirable model in terms of F_1. Given the success of conditionally trained models for the toy domain discussed above and the apparent correlation of CLL and F_1 on real data sets, an obvious suggestion is to train the flexible and fixed HMMs to directly maximize conditional likelihood.

3.2 Optimizing conditional likelihood (CG)

EM is a convenient method for maximizing joint likelihood. While iterative lower bounding techniques for conditional likelihood also exist (Jebara and Pentland 1998), it turns out that for our problem that the form of our objective function is well suited for generic nonlinear optimization techniques such as conjugate gradient descent (CG). Our derivation of the objective value and its derivatives is similar to Krogh (94). We sketch only an outline here.[2]

Our objective function *P(C|W)* can be written as a likelihood ratio:

$$P(\vec{C} \mid \vec{W}) = \frac{P(\vec{C},\vec{W})}{\sum_{C'} P(\vec{C}',\vec{W})}$$

In our model, we can obtain these quantities from summing out S:

$$\frac{P(\vec{C},\vec{W})}{\sum_{C'} P(\vec{C}',\vec{W})} = \frac{\sum_{S} P(\vec{C},\vec{S},\vec{W})}{\sum_{S}\sum_{C'} P(\vec{C}',\vec{S},\vec{W})}$$

The numerator is the same constrained likelihood we computed in the last section for use with Baum-Welch, and the denominator is the identical quantity, only without the class constraints. These quantities can thus be efficiently computed by using the forward-backward algorithm twice (in parallel, for greatest efficiency). In practice, for numerical stability and mathematical simplicity, we actually optimize log $P(C|W)$ and we consider our model parameters to be not the actual transition and emission probabilities but rather the logs of these quantities.

In order to efficiently optimize this objective, we would like to know not only its value, but also the partial derivatives with respect to each model parameter (transitions and emissions). The derivatives have a simple, intuitive form, which we give here for emission parameters *P(w|s)*; the transitions are identical.

$$\frac{\partial \, Log \, P(\vec{C} \mid \vec{W})}{\partial \, Log \, P(w \mid s)} = \varepsilon_c[s \to w] - \varepsilon_u[s \to w]$$

Where ε_c is the conditional expectation of the emission w in state s given the class constraints and ε_u is its expectation ignoring the class constraints. These quantities can be calculated in the process of computing the value function above.[3]

Note that the log-parameters returned by CG will each be arbitrary real numbers, meaning that in general the HMM will not represent a probability distribution; moreover if globally normalized to represent one, it will generally be radically deficient. This does not cause problems for computing Viterbi sequences, however, which is all that one needs for classification.

Given the same parameter initializations we use with EM (slightly perturbed uniform transitions, corpus-averaged unigram emissions), both the flexible and fixed HMMs consistently converge to local maxima of conditional likelihood that yield 0.0 F_1 on test data. This is at first surprising given the high performance on toy data. Comparing the parameters of the joint and conditional HMMs, we see that the transition weights are qualitatively similar, but that the emission weights remain much closer to uniform in the conditional model than in the joint model, and vary much less from state to state. Thus, the probability of generating target words in the target state is not significantly higher than generating them in the background state.

[2] A detailed derivation is available in an online appendix.

[3] We also use a weak gaussian prior for regularization of parameters and a sum-constraint over all parameters to remove a spurious degree of freedom in optimization.

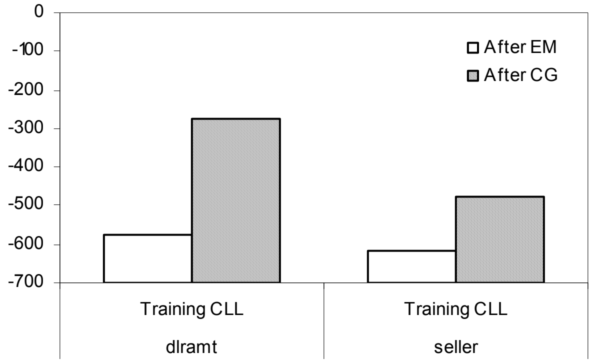

Figure 6. CG run on output of EM improves CLL.

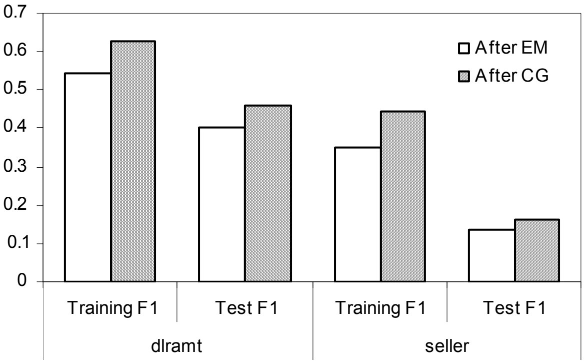

Figure 7. CG run on output of EM improves F1.

Since moving to the target state requires following low-probability transitions out of the background (and through a prefix in the fixed structure), the most probable state sequence is to constantly remain in the background, yielding no guesses at targets. This is not to say the conditional likelihood of being in a target state over actual targets is necessarily unreasonable, just that it is no larger than the probability of being in a non-target state.

It is worth realizing that this is partly because company names and dollar amounts do appear in the background with relatively high frequency. In fact, even the highest probability target emissions for *dlramt* or a company class like *purchaser* do not have sufficient discriminative power on their own to suggest emitting them from a target state. Tables 3 and 4 provide some examples of common target emissions and their generative and discriminative power. For reference, the corpus has a total of 81,288 words of which 715 are labeled as *dlramt* and 1,885 are labeled as *purchaser*.

	Generative		Discriminative	
word	*P(w|t)*	*P(w|~t)*	*P(t|w)*	*P(~t|w)*
mln	0.193	0.005	0.236	0.764
dlrs	0.188	0.007	0.199	0.801

Table 3. Common emissions for *dlramt*

	Generative		Discriminative	
word	*P(w|t)*	*P(w|~t)*	*P(t|w)*	*P(~t|w)*
Corp	0.071	0.003	0.338	0.662
General	0.011	0.0003	0.467	0.533
Inc	0.084	0.005	0.306	0.694
International	0.012	0.0006	0.239	0.707

Table 4. Common emissions for *purchaser*

3.3 The conditional search space problem

The failure of the conditionally trained model to adequately differentiate states could either be a search problem or a shortcoming of using conditional likelihood as an objective. However, it clearly seems to be a search problem. We can see that the models trained with EM not only perform better (in F_1), but have higher conditional likelihood than the conditionally trained models. We can also see from Figure 5 above that conditional likelihood is well correlated with F_1. Even though CG will find a local optimum in conditional likelihood, we have the counter-intuitive result that EM is in practice a better optimizer of conditional likelihood than CG, despite not being even locally optimal. Since CG is locally optimal, the task becomes one of finding a better initial parameter setting from which to run it.

Given that the joint models have higher conditional likelihood than the conditional models, an obvious choice is to use HMMs trained with EM as the input to CG to maximize conditional likelihood. As Figures 6 and 7 show, running CG after EM consistently increased both CLL and F_1. In fact, running CG after even a single round of results in comparable performance gains. These results demonstrate that EM is finding a relatively promising basin for the conditional likelihood optimization, but is not finding a local maximum (nor would it be expected to do so).

4 Fixed vs. flexible transition structures

Theoretically flexible structures as a search space subsume structures in which some of the parameters are fixed (e.g., certain transitions are fixed at

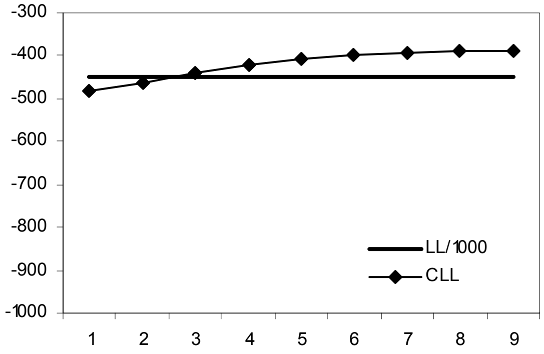

Figure 8. EM on fixed structure yields higher CL.

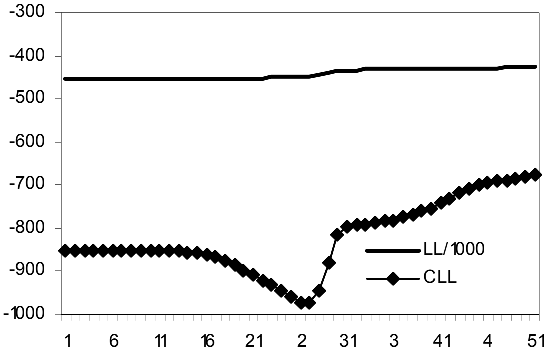

Figure 9. EM on flexible structure yields lower CL.

0). Nevertheless, it has been observed in many cases that using (clever) fixed structures often results in better F_1 and in states whose roles are intuitively more like what people expect they should be (Freitag and McCallum 2000).

A reason is that by fixing the structure, the hidden state sequence becomes less hidden. This means that EM (or even CG) has less work to do, and fewer degrees of freedom, because there is less room to consider different state sequences. Consider the effect of removing the self-transitions from the prefix and suffix state of the fixed structure—in this case, the state sequence would be fully determined, because all background emissions must come from the background state with the exception of the word immediately preceding and immediately following the target. These must be generated by the prefix and suffix state respectively, leaving all the target emissions for the target state. In this case, there is no hidden structure and the search space has a unique maximizer for both joint and conditional likelihood. As we increase the flexibility of the possible state sequences (either by adding self-transitions or by adding more states), the search space becomes more complex, and the optimization procedures must begin to define the roles of the states in some manner.

However, in the fixed structure with self-transitions, the only flexibility in the search space is how early to move into the prefix state and how late to move out of the suffix state. This is reflected by the relative strength of those self-transition probabilities, which is all that qualitatively changes across successive iterations of EM.

In such a constrained space, optimizing joint likelihood with EM also does an impressive job of maximizing conditional likelihood.[4] As Figures 8 and 9 show, EM does a better job maximizing conditional likelihood for fixed structures than for flexible structures. Thus we see that human intuition about discriminative roles for HMM states in information extraction systems is validated by the empirical result that this subspace of models has higher conditional likelihood than a uniform point in the larger space of more flexible models.

5 Increasing model and task complexity

The experiments presented above were conducted using small HMMs that can easily be studied and understood. For robust performance, a larger number of target and background states is usually required. When training a larger HMM, it becomes clearer that one of EM's primary effects is to cluster the background emissions into distributional clusters and link them to model flat syntactic patterns. For example, Table 5 shows a few selected states from an HMM trained with 7 background states and 4 target states. The first two states are background states and the last two are *dlramt* states. Clearly state 5 has been specialized to generate prepositions and state 3 has been specialized to produce nouns, specifically amounts of currency (broadly defined). Similarly state 9 emits numbers and state 12 emits number magnitudes. The transitions reflect common patterns among these states—*P(S3|S5)* = 0.74 and *P(S12|S9)* = 0.94. This models PPs and compound numbers respectively.

[4] Of course, the benefit isn't just from any constraint, but from good constraint.

State 3	State 5	State 9	State 12
shares	of	Two	Mln
stock	in	240	Billion
dlrs	by	757	MLN
offer	with	985	Purchase
pct	for	about	a

Table 5. EM clusters emissions to differentiate states.

Once emissions have been clustered, the states take on specific roles and the transition structure quickly becomes determined. Thus when models trained by EM are further optimized for conditional likelihood, CG can recruit these specialized states as needed for classification. In contrast, when training CG from the beginning, the meaningful transitions (from background to target and back) are few and far between since CG is uninterested in internal background structure. Thus the roles for individual background states are less clear and differentiating them is difficult (as we saw in section 3.2).

When running CG after EM on more difficult targets with larger models, the behavior is not always as clean as that shown above. Conditional likelihood is still consistently boosted, but the correlation between training CL and test F_1 is not always as strong. In future research we will continue analyze more complex HMMs and look for better techniques for finding promising basins in which to maximize conditional likelihood.

6 Conclusion

Ideally, one would maximize test set F_1 for IE. The closest one can get to this is to maximize training F_1, which is generally only possible for discrete search, such as structure search. We have shown that conditional likelihood is better correlated with F_1 than joint likelihood is. For simple enough examples, this can be usefully maximized directly. However, an ironic result is that EM can sometimes be the best available tool for the broad maximization of CL and F_1. This is partially because, whatever else it does right or wrong, EM naturally acts as a distributional clustering tool (Rooth et al. 1999; Clark 2000). To the extent that having states represent distributional word classes is better than having them represent nothing at all, EM is a useful first step. Improved results can then be gained in two ways. First, by maximizing CL starting from the output of EM, since CG maximization is a good tool for local improvement of CL. Second, by constraining structure, we can force EM to give parameters with better CL and F_1 than it would otherwise produce.

This paper has developed on developing theoretical understanding of the empirical successes and failures of HMMs for information extraction trained to maximize joint likelihood by EM, and trained to maximize conditional likelihood. Future work will emphasize making use of this understanding in large-scale applications.

References

Appelt, D., Hobbs, J., Bear, J., Israel, D. J. and Tyson, M. FASTUS: A Finite-State Processor for Information Extraction from Real-World Text, in *Proceedings of IJCAI-93*, Chambery, France, Sep 1993.

Bahl et al. [1986] Maximum mutual information estimation of hidden markov model parameters for speech recognition. In Proceedings of ICASSP 1986, 49-52.

D. Bikel, S. Miller, R. Schwartz, R. Weischedel (1997), "NYMBLE: A High-Performance Learning Name Finder." Proc. Applied Natural Language Processing, 1997.

Thorsten Brants, 2000. TnT - A Statistical Part-of-Speech Tagger. In Proceedings of the Sixth Applied Natural Language Processing Conference ANLP-2000, Seattle, WA.

Alexander Clark (2000) Inducing Syntactic Categories by Context Distribution Clustering, Proceedings of CoNLL 2000, September 2000, Lisbon.

Durbin, R., Eddy, S., Krogh, A. and Mitchison, G. 1998. *Biological sequence analysis: Probabilistic models of proteins and nucleic acids.* Cambridge University Press.

Freitag, Dayne. 1998. Machine Learning for Information Extraction in Informal Domains. PhD thesis, Carnegie Mellon. Technical report CMU-CS-99-104.

Freitag, D., & McCallum, A. (2000). Information extraction with HMM structures learned by stochastic optimization. *Proceedings of the Eighteenth Conference on Artificial Intelligence (AAAI-2000).*

Jebara T. and Pentland A. (1998). Maximum conditional likelihood via bound maximization and the CEM algorithm. NIPS 11

Juang and Rabiner [1991] Hidden Markov models for speech recognition. Technometrics 33:251-272.

Krogh [1994] Hidden Markov models for labeled sequences. Proceedings of the 12th IAPR International Conference on Pattern Recognition 140-144. IEEE.

N. Kushmerick, D. Weld and R. Doorenbos. Wrapper induction for information extraction, IJCAI-97, 1997

T. R. Leek. Information extraction using hidden Markov models. Master's thesis, UC San Diego, 1997

D. Lewis , Representation and Learning in Information Retrieval, Ph.D. dissertation, University of Massachusetts, 1992

Normandin and Morgera [1991] An improved MMIE training algorithm for speaker-independent, small vocabulary, continuous speech recognition. Proceedings of ICASSP 1991, 537-540

L.R. Rabiner, A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, Proceedings of IEEE, 1989, 257-286.

Mats Rooth, Stefan Riezler, Detlef Prescher, Glenn Carroll, and Franz Beil. Inducing a Semantically Annotated Lexicon via EM-Based Clustering Proceedings of ACL '99, pp. 104--111.