Information Extraction with HMMs and Shrinkage

Dayne Freitag
Just Research
Pittsburgh, PA 15213
dayne@justresearch.com

Andrew Kachites McCallum
Just Research
Pittsburgh, PA 15213
mccallum@justresearch.com

Abstract

"Information extraction" refers to the process of converting text documents to structured content summaries. Such summaries can be presented to users or be used by software agents engaged in text mining. This paper advocates the use of HMMs for information extraction. The HMM state transition probabilities and word emission probabilities are learned from labeled training data. As in many learning problems, however, the lack of sufficient labeled training data hinders the reliability of the model. The key contribution of this paper is the use of relationships between HMM states and a statistical technique called "shrinkage" in order to significantly improve estimation of the HMM emission probabilities in the face of sparse training data. In experiments on seminar announcements and Reuters acquisitions articles, shrinkage is shown to reduce error by up to 40%, and the resulting HMM outperforms a state-of-the-art rule-learning system.

1 Introduction

The Internet makes available a tremendous amount of text that has been generated for human consumption; unfortunately, this information is not easily manipulated or analyzed by computers. *Information extraction* is the process of filling fields in a database by automatically extracting fragments of human-readable text. Examples include extracting the location of a meeting from an email message, or the name of the acquired company from a news article about a company takeover.

This paper advocates the use of hidden Markov models (HMMs) for information extraction. HMMs have been applied with significant success to many language-related tasks, including part-of-speech tagging [Kupiec, 1992] and speech recognition [Rabiner, 1989]. Because HMMs have foundations in statistical theory, there is a rich body of established techniques for learning the parameters of an HMM from training data and for classifying test data. In our work HMM state-transition probabilities

and word emission probabilities are learned from labeled training data. However, as in many learning problems, large amounts of training data are required to learn a model that generalizes well. Since training data must usually be painstakingly labeled by hand, it is often difficult to obtain enough, and the small quantities of available training data limit the performance of the learned extractor. (When learning tasks in which the output is the identity of the hidden states, several studies have shown that incorporating unlabeled data with Baum-Welch only degrades performance [Kupiec, 1992; Seymore, McCallum, & Rosenfeld, 1999].)

The key contribution of this paper is the integration of a statistical technique called *shrinkage* into information extraction by HMMs. In our approach, shrinkage is used to learn more robust HMM emission probabilities in the face of limited training data. The technique works by "shrinking" parameter estimates in data-sparse individual states towards the estimates calculated for data-rich conglomerations of states in ways that are provably optimal under the appropriate conditions. Shrinkage has been widely used in statistics and language modeling, including in HMMs for acoustic modeling in speech recognition [Lee, 1989].

In our approach to information extraction, the HMM forms a probabilistic generative model of an entire document from which sub-segments are to be extracted. In each HMM, a subset of the states are distinguished as "target" states, and any words of the document that are determined to have been generated by those states are part of the extracted sub-sequence.

We describe experiments on two real-world data sets: on-line seminar announcements and Reuters newswire articles on company acquisitions. Results show that shrinkage consistently improves the performance over smoothing by absolute discounting. The HMM also out-performs a state-of-the-art rule-learning system.

2 HMMs for Information Extraction

A HMM is a finite state automaton with stochastic state transitions and word emissions [Rabiner, 1989]. Associated with each of a set of states, $S = \{s_1, \cdots, s_n\}$, is a probability distribution over the words in the emission vocabulary $V = \{w_1, \cdots, w_n\}$. The probability that state s_j will emit word $w \in V$ is written $\mathrm{P}(w|s_j)$. Similarly, the probability of moving from state s_i to state s_j is written $\mathrm{P}(s_j|s_i)$.

The models we use for information extraction have the following four characteristics: **(1)** Each HMM extracts just one type of field (such as "purchasing price"). When multiple fields are to be extracted from the same document (such as "purchasing price" and "acquiring company"), a separate HMM is constructed for each field. **(2)** They model the entire document, and thus do not require pre-processing to segment document into sentences or other pieces. **(3)** They contain two kinds of states, background states and target states. Target states are responsible for the text to be extracted. **(4)** They are not fully connected. The restricted transition structure, which we construct manually, captures context that helps improve extraction accuracy. Given a model and all its parameters, information extraction is performed by using the Viterbi algorithm to determine the sequence of states that was most likely to have generated the entire document, and extracting the words that were associated with designated "target" states.

Figure 1 shows three example topologies. The model in Figure 1(a) is the simplest

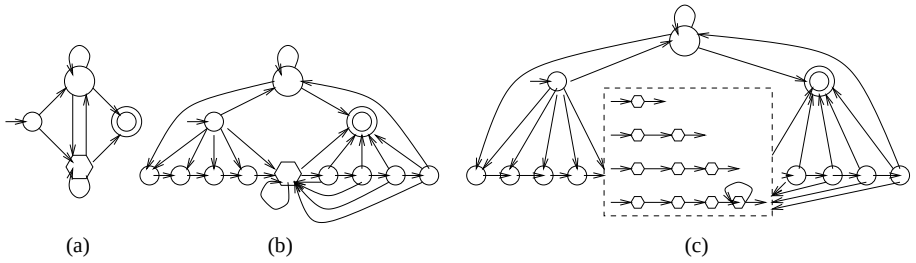

Figure 1: Three example topologies. Source-less arrows indicate start state; double circles indicate terminal state; hexagons indicate target states.

possible topology. In practice, we expect that context around the target state to provide important clues in the search for target text. We can exploit some of these clues by adding prefix and suffix states, as in Figure 1(b). Similarly, target fragments can vary in length, and certain tokens may be more common at the beginning or end of the fragments. If the object is to extract the name of a company for example, the tokens "Inc" and "Corp" are almost certainly at the end of a fragment. We can attempt to capture such structure by expanding the single target state into an array of parallel paths of varying length. Figure 1(c) shows a set of target paths of lengths one to four. In order to train a model, each token in a document is labeled according to whether it is part of the target text. We require that only target states emit such tokens, and only non-target states emit non-target tokens.

When the emission vocabulary is large with respect to the number of training examples, maximum likelihood estimation of emission probabilities will lead to poor estimates, with many words inappropriately having zero probability. This can be prevented by, for example, Bayes optimal parameter estimation in conjunction with a uniform Dirichlet prior (*Laplace smoothing*). An alternative smoothing technique that we have found to perform better when the number of zero-count words varies widely from state to state is *absolute discounting*, commonly used in statistical language modeling for speech recognition. (More complex methods like Good-Turing fail with the extremely small per-state training sets we have.)

Both Laplace smoothing and absolute discounting calculate the word distribution in a state using only the training data in the state itself. In the next section, we discuss *shrinkage*, a method that leverages the word distributions in several related states in order to improve parameter estimation.

3 Shrinkage

In many machine learning tasks there is a tension between constructing complex models with many states and constructing simple models with only a few states. A complex model is able to represent intricate structure of a task, but often results in poor (high variance) parameter estimation because the training data is highly fragmented. A simple model results in robust parameter estimates, but performs poorly because it is not sufficiently expressive to model the data (too much bias).

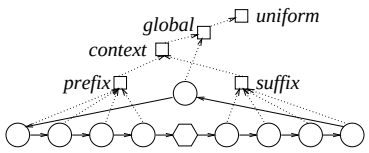

Figure 2: A shrinkage configuration that addresses data sparsity in contextual states, showing shrinkage only for *non-target* states.

Shrinkage is a technique that balances these competing concerns by "shrinking" parameter estimates from data-sparse states of the complex model toward the estimates in related data-rich states of simpler models. The combination of estimates is provably optimal under the appropriate conditions. Shrinkage has been extensively studied in statistics [Carlin & Louis, 1996]. We employ a simple form of shrinkage that combines the estimates in a weighted average, and learns the weights with EM. In speech recognition this is called *deleted interpolation* [Jelinek & Mercer, 1980].

Shrinkage for HMMs and Information Extraction. Shrinkage is typically defined in terms of a hierarchy representing the expected similarity between parameter estimates, with the estimates at the leaves. We create such a hierarchy by defining subsets of states that have word emission distributions we expect to be similar, and declare them to share a common "parent" in a hierarchy of word distributions.

Figure 2 shows such a hierarchy. It depicts, for example, that all prefix states are expected to have related word distributions—reflecting also the fact that in a simpler model, all four prefix states might have been represented by a single state that allowed up to four self-transitions. Internal nodes of the hierarchy can also have parents, reflecting expectations about weaker similarity between groups of states, and representing HMM emission distributions that are yet again more simple. At the top of each hierarchy is the most unassuming of all word distributions, the uniform distribution, which gives all words in the vocabulary equal probability. Because the uniform distribution is included we no longer need to smooth the local estimates with Laplace or absolute discounting.

We have compared several shrinkage hierarchy configurations. Given that we distinguish four classes of states—*non-target*, *target*, *prefix*, and *suffix*—the four shrinkage configurations are as follows: **None:** No shrinkage; only absolute discounting is used. **Uniform:** Instead of absolute discounting, all single-state distributions are shrunk toward the uniform distribution. **Global:** The distributions of all *target* states are shrunk toward a common parent, as well as the uniform distribution; likewise for the *non-target* states with a different parent. **Hierarchical:** (Shown in Figure 2.) *Target* distributions are handled in the same way as in *global*. Each of the other classes of states—*non-target*, *prefix*, and *suffix*—is shrunk toward a separate, class-specific parent. The prefix and suffix parents are furthermore shrunk toward a shared "context" grandparent. Finally, all *non-target*, *prefix*, and *suffix* states are also shrunk toward a single ancestor, shared among all states that are not *target* states. Again, every state is also shrunk toward the uniform distribution.

Shrinkage-Based Word Probabilities. Our new, shrinkage-based parameter estimate in a leaf of the hierarchy (state of the HMM) is a linear interpolation

of the estimates in all distributions from the leaf to its root. Local estimates are calculated from their training data by maximum likelihood (simple ratios of counts, with no additions or discounting). The training data for an internal node of the hierarchy is the union of all the data in its children.

We write the word probability estimates for the nodes on the path starting at state s_j as $\{\mathrm{P}(w|s_j^0), \mathrm{P}(w|s_j^1), ...\mathrm{P}(w|s_j^k)\}$, where $\mathrm{P}(w|s_j^0)$ is the estimate at the leaf, and $\mathrm{P}(w|s_j^k)$ is the uniform distribution at the root. The interpolation weights among these estimates are written $\{\lambda_j^0, \lambda_j^1, ...\lambda_j^k\}$, where $\sum_{i=1}^{k} \lambda_j^i = 1$. The new, shrinkage-based estimate for the probability of word w in state s_j is written $\hat{\mathrm{P}}(w|s_j)$, and is the weighted average: $\hat{\mathrm{P}}(w|s_j) = \sum_{i=1}^{k} \lambda_j^i \mathrm{P}(w|s_j^i)$.

Determining Mixture Weights. We derive empirically optimal weights, λ_j^i, among the ancestors of state s_j, by finding the weights that maximize the likelihood of some hitherto unseen "held-out" data, $\mathcal{H}$. This maximum can be found by a form of Expectation-Maximization (EM), where each word is assumed to have been generated by first choosing one of the hierarchy nodes in the path to the root, say s_j^i (with probability λ_j^i), then using that node's word distribution to generate that word. EM then maximizes the total likelihood when the choices of nodes made for the various words are unknown. EM begins by initializing the λ_j's to some initial values, say $\lambda_j^i = \frac{1}{k}$, then iterating the E- and M-steps until the λ_j's do not change. In the **E-step**, we calculate, β_j^i, the degree to which each ancestor i predicts the words in state s_j's held-out set, $\mathcal{H}_j$. In the **M-step** we derive new (and guaranteed improved) weights by normalizing the β's:

$$\textbf{E-step:} \quad \beta_j^i = \sum_{w_t \in \mathcal{H}_j} \frac{\lambda_j^i \mathrm{P}(w_t|s_j^i)}{\sum_m \lambda_j^m \mathrm{P}(w_t|s_j^m)} \qquad \textbf{M-step:} \quad \lambda_j^i = \frac{\beta_j^i}{\sum_m \beta_j^m} \tag{1}$$

While correct and conceptually simple, this method makes inefficient use of the available training data by carving off a held-out set. We fix this problem by evaluating the E-step with each individual word occurrence held out in turn. This method is very similar to the "leave-one-out" cross-validation commonly used in statistical estimation.

4 Experiments

We present experimental results on nine information extraction problems from two corpora: a collection (485 documents) of seminar announcements posted to local newsgroups at a large university, and a collection (600 documents) of articles describing corporate acquisitions taken from the Reuters dataset [Lewis, 1992]. Both of these datasets, as well as the IE problems defined for them, are described in detail in previously published work [Freitag, 1999]. For each problem we report performance averaged over five random splits of the corresponding corpus into training and testing sets of equal size. Let N be the number of test documents that contain a target fragment, M the number of documents for which the learner predicts a target fragment, and C the number of these predictions that exactly identify a target fragment. *Precision* (P) is C/M, and *recall* (R) is C/N. We measure performance in terms of *F1*, a metric common in information retrieval, which is the harmonic mean of P and R, i.e. $2/(1/P + 1/R)$.

	Context	Paths	Shrinkage	*speaker*	*location*	*stime*	*etime*
1.	1	1	None	0.431	0.797	0.943	0.771
2.	10	1	None	0.363	0.558	0.967	0.746
3.	4	1	None	0.460	0.653	0.960	0.716
4.	4	1	Uniform	0.499	0.660	0.971	0.840
5.	4	1	Global	0.558	0.758	0.984	0.589
6.	4	1	Hier.	0.531	0.695	0.976	0.565
7.	4	4	None	0.513	0.735	0.991	0.814
8.	4	4	Uniform	0.614	0.776	0.991	0.933
9.	4	4	Global	0.711	0.839	0.991	0.595
10.	4	4	Hier.	0.672	0.850	0.987	0.584

Table 1: F1 performance with various topologies and shrinkage configurations.

The performance of an algorithm is measured document by document. If the task is to extract the start time of a seminar from an announcement, we assume that there is a single correct answer (perhaps presented several different times in the same or superficially different ways). We ask whether a learner's single best prediction exactly identifies one of the fragments representing the start time. If a learner's best prediction does not align exactly with an actual start time, as identified by the human labeler, it is counted as an error.

Table 1 presents the results of experiments with ten different model topologies and shrinkage configurations. Two trends deserve particular notice. First, in general, performance increases with context size and, especially, path count. Compare, for example, Rows 1, 3, and 7, all of which use absolute discounting instead of shrinkage. Note, however, that among the larger models using absolute discounting none performs better than the simplest model (Row 1) on all four tasks. In other words, model elaboration often degrades performance, presumably because of data sparsity. The second salient trend is that shrinkage clearly ameliorates data sparsity. Simply shrinking state emission estimates toward the uniform distribution appears superior to absolute discounting. With the exception of one of the four tasks (*etime*), "Global" shrinkage (Rows 5 and 9) leads to the best performance. On *speaker*, global shrinkage reduces the error of the corresponding no-shrinkage model by 40%.

We attribute the large performance differences on the *etime* task to the relative infrequency of this field; it appears in only about half of the documents. The decreased *etime* performance in Rows 5, 6, 9, and 10 is due to a large number of predictions for documents in which *etime* does not appear. Observing that *stime* and *etime* tend to occur in close proximity, we experimented with a model designed and trained to extract *stime* and *etime* at the same time; the model had two sets of context and target states, one for each field. Using this model along with global shrinkage, we observed an F1 performance for *etime* of 0.849 which, while not the best across all models, nevertheless improves upon the models using absolute discounting.

It is interesting to ask how the distribution of mixture weights varies as a function of a state's role in the model. Table 2 shows, for sample runs on each of the four seminar announcement tasks, how much weight is placed on the local token distribution of each of four prefix states. The "global" shrinkage configuration is used in this case. Note how the local weight tends to decline with increasing

Distance	*speaker*	*location*	*stime*	*etime*
1	0.84	0.84	0.92	0.95
2	0.81	0.90	0.98	0.98
3	0.73	0.80	0.85	0.95
4	0.65	0.74	0.86	0.93

Table 2: Local mixture weights along the prefix path as a function of distance from the target states.

	speak.	*loc.*	*stime*	*etime*	*acq*	*purch*	*acqabr*	*dlramt*	*status*
SRV	0.703	0.723	0.988	0.839	0.343	0.429	0.351	0.527	0.380
HMM	0.711	0.839	0.991	0.595	0.309	0.481	0.401	0.553	0.467

Table 3: F1 of SRV and a representative HMM on nine fields from two domains, the seminar announcements and corporate acquisitions.

distance from the target text, agreeing with our intuition that the most consistent patterns are the closest. Also, local weight decreases in proportion to the difficulty of the field, as reflected in F1 results. Clearly, the two time fields tend to occur in very predictable contexts.

Table 3 compares performance of a fixed model (the one listed in Row 9 of Table 1) on nine information extraction problems with the performance of SRV, a consistently strong rule-learning algorithm described elsewhere [Freitag, 1999]. On all but *etime* and *acq*, the HMM obtains a higher F1 score than SRV. Note, however, that it is a simple matter to design a model that outperforms SRV on the *etime* task (as described in the previous paragraph or shown in Row 8 of Table 1).

5 Related Work and Conclusions

HMMs have been previously applied to information extraction by methods that differ from our approach in various ways. Seymore, McCallum, & Rosenfeld present an effort to learn HMMs state/transition structure [1999]. Unlike this paper, the approach uses a single HMM to extract many fields which are densely packed in moderately structured text (such as research paper references and headers). Leek applies HMMs to the problem of extracting gene locations from biomedical texts [Leek, 1997]. In contrast with the models we study, Leek's models are carefully hand-engineered for the task—both the general topology (which is hierarchical and complex), and the language models of individual states. The Nymble system [Bikel *et al.*, 1997] uses HMMs to perform "named entity" extraction as defined by MUC-6. All different fields to be extracted are modeled in a single HMM, but to avoid the resulting difficult structure-learning problem, there is a single state per target and the state-transition structure is completely connected.

This paper has demonstrated the ability of shrinkage to improve the performance of HMMs for information extraction. The tension between the desire for complex models and the lack of training data is a constant struggle here (as in many machine learning tasks), and shrinkage provides a principled method of striking a balance.

References

[Bikel *et al.*, 1997] Bikel, D. M.; Miller, S.; Schwartz, R.; and Weischedel, R. 1997. Nymble: a high-performance learning name-finder. In *Proceedings of ANLP-97*, 194–201.

[Carlin & Louis, 1996] Carlin, B., and Louis, T. 1996. *Bayes and Empirical Bayes Methods for Data Analysis*. Chapman and Hall.

[Dempster, Laird, & Rubin, 1977] Dempster, A.; Laird, N.; and Rubin, D. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society* 39(B):1–38.

[Freitag, 1999] Freitag, D. 1999. *Machine Learning for Information Extraction in Informal Domains*. Ph.D. Dissertation, Carnegie Melon University.

[Jelinek & Mercer, 1980] Jelinek, F., and Mercer, R. 1980. Interpolated estimation of Markov source parameters from sparse data. In Gelsema, S., and Kanal, L. N., eds., *Pattern Recognition in Practice*, 381–402.

[Kupiec, 1992] Kupiec, J. 1992. Robust part-of-speech tagging using a hidden Markov model. *Computer Speech and Language* 6:225–242.

[Lee, 1989] Lee, K.-F. 1989. *Automatic Speech Recognition: The Development of the SPHINX System*. Kluwer Academic Publishers.

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

[Lewis, 1992] Lewis, D. 1992. *Representation and Learning in Information Retrieval*. Ph.D. Dissertation, Univ. of Massachusetts. CS Tech. Report 91–93.

[Rabiner, 1989] Rabiner, L. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE* 77(2).

[Seymore, McCallum, & Rosenfeld, 1999] Seymore, K.; McCallum, A.; and Rosenfeld, R. 1999. Learning hidden markov model structure for information extraction. Submitted to the AAAI-99 Workshop on Machine Learning for Information Extraction.

[van Mulbregt *et al.*, 1998] van Mulbregt, P.; Carp, I.; Gillick, L.; and Yamron, J. 1998. Text segmentation and topic tracking on broadcast news via a hidden markov model approach. In *Proceedings of the International Conference on Spoken Language Processing*.