Hierarchical Hidden Markov Models for Information Extraction

Marios Skounakis[+t]
marios@cs.wisc.edu

Mark Craven**
craven@biostat.wisc.edu

Soumya Ray**
sray@cs.wisc.edu

* Department of Computer Sciences
University of Wisconsin
Madison, Wisconsin 53706

Department of Biostatistics & Medical Informatics
University of Wisconsin
Madison, Wisconsin 53706

Abstract

Information extraction can be defined as the task of automatically extracting instances of specified classes or relations from text. We consider the case of using machine learning methods to induce models for extracting relation instances from biomedical articles. We propose and evaluate an approach that is based on using *hierarchical* hidden Markov models to represent the grammatical structure of the sentences being processed. Our approach first uses a shallow parser to construct a multi-level representation of each sentence being processed. Then we train hierarchical HMMs to capture the regularities of the parses for both positive and negative sentences. We evaluate our method by inducing models to extract binary relations in three biomedical domains. Our experiments indicate that our approach results in more accurate models than several baseline HMM approaches.

1 Introduction

In many application domains, there is the potential to greatly increase the utility of on-line text sources by using automated methods for mapping selected parts of the unstructured text into a structured representation. For example, the curators of genome databases would like to have tools that could accurately extract information from the scientific literature about entities such as genes, proteins, cells, diseases, etc. For this reason, there has been much recent interest in developing methods for the task of *information extraction* (IE), which can be defined as automatically recognizing and extracting instances of specific classes of entities and relationships among entities from text sources.

Machine learning methods often play a key role in IE systems because it is difficult and costly to manually encode the necessary extraction models. Hidden Markov models (HMMs) [Leek, 1997; Bikel *et al,* 1999; Freitag and McCallum, 2000], and related probabilistic sequence models [McCallum *et at,* 2000; Lafferty *et al.,* 2001], have been among the most accurate methods for learning information extractors. Most of the work in learning HMMs for information extraction has focused on tasks with semi-structured and other text sources in which English grammar does not play a key role. In contrast, the task we consider here is extracting information from abstracts of biological articles [Hirschman *et* a/., 2002]. In this domain, it is important that the learned models are able to represent regularities in the grammatical structure of sentences.

Here we report the identification of an integral membrane ubiquitin-conjugating enzyme. This enzyme, UBC6, localizes to the endoplasmic reticulum, with the catalytic domain facing the cytosol.

subcellular-localization(UBC6,endoplasmic reticulum)

Figure 1: An example of the information extraction task. The top of the figure shows part of a document from which we wish to extract instances of the SUbcellular-localization relation. The bottom of the figure shows the extracted tuple.

In this paper, we present an approach based on using *hierarchical hidden Markov models* (HHMMs) [Fine *etal,* 1998] to extract information from the scientific literature. Hierarchical hidden Markov models have multiple "levels" of states which describe input sequences at different levels of granularity. In our models, the top level of the HMMs represent sentences at the level of phrases, and the lower level of the HMMs represent sentences at the level of individual words. Our approach involves computing a shallow parse of each sentence to be processed. During training and testing, the hierarchical HMMs manipulate a two-level description of the sentence parse, instead ofjust processing the sentence words directly. We evaluate our approach by extracting instances of three binary relations from abstracts of scientific articles. Our experiments show that our approach results in more accurate models than several baseline approaches using HMMs.

An example of a binary relation that we consider in our experiments is the subcellular-localization relation, which represents the location of a particular protein within a cell. We refer to the *domains* of this relation as PROTEIN and LOCATION. We refer to an instance of a relation as a *tuple.* Figure 1 provides an illustration of our extraction task. The top of the figure shows two sentences in an abstract, and the bottom of the figure shows the instance of the target relation

subcellular-localization that we would like to extract from the second sentence. This tuple asserts that the protein UBC6 is found in the subcellular compartment called the endoplasmic reticulum. In order to learn models to perform this task, we use training examples consisting of passages of text, annotated with the tuples that should be extracted from them.

In earlier work [Ray and Craven, 2001], we presented an approach that incorporates grammatical information into single-level HMMs. The approach described in this paper extends the earlier work by using hierarchical HMMs to provide a richer description of the information available from a sentence parse.

Hierarchical HMMs originally were developed by Fine *et al.* (1998), but the application of these models to information extraction is novel, and our approach incorporates several extensions to these models to tailor them to our task. Bikel *el al.* (1999) developed an approach to *named enlily* recognition that uses HMMs with a multi-level representation similar to a hierarchical HMM. In their models, the top level represents the classes of interest (e.g. person name), and the bottom level represents the words in a sentence being processed. Our approach differs from theirs in several key respects: (i) our input representation for all sentences being processed is hierarchical, (ii) our models represent the shallow phrase structure of sentences, (iii) we focus on learning to extract *relations* rather than entities, (iv) we use *null* models to represent sentences that do not describe relations of interest, and (v) we use a discriminative training procedure. Miller *el al.* (2000) developed an information-extraction approach that uses a lexicalized, probabilistic context-free grammar (LPCFG) to simultaneously do syntactic parsing and semantic information extraction. The genre of text that we consider here, however, is quite different from the news story corpus on which available LPCFGs have been trained. Thus it is not clear how well this intriguing approach would transfer to our task.

2 Sentence Representation

In most previous work on HMMs for natural language tasks, the passages of text being processed have been represented as sequences of tokens. A hypothesis underlying our work is that incorporating sentence structure into the learned models will provide better extraction accuracy. Our approach is based on using syntactic parses of all sentences to be processed. In particular, we use the Sundance system [Riloff, 1998] to obtain a shallow parse of each given sentence.

The representation we use in this paper does not incorporate all of the information provided by the Sundance parser. Instead our representation provides a partially "flattened", two-level description of each Sundance parse tree. The top level represents each sentence as a sequence of phrase segments. The lower level represents individual tokens, along with their part-of-speech (POS) tags. In positive training examples, if a segment contains a word or words that belong to a domain in a target tuple, the segment and the words of interest are annotated with the corresponding domain. We iefer to these annotations as *labels.* Test instances do not contain labels - the labels are to be predicted by the learned IE model.

Figure 2 shows a sentence containing an instance of the

"This enzyme, UBC6, localizes to the endoplasmic reticulum, with the catalytic domain facing the cytosol."

	(a)	(b)	(c)
1	NP_SEGMENT	DET UNK	this enzyme
2	NP_SEGMENT:PROTEIN	UNK:PROTEIN	ubc6
3	VP_SEGMENT	V	localizes
4	PP_SEGMENT	PREP	to
5	NP_SEGMENT:LOCATION	ART N LOCATION N:LOCATION	the endoplasmic reticulum
6	PP_SEGMENT	PREP	with
7	NP_SEGMENT	ART N UNK	the catalytic domain
8	VP_SEGMENT	V	facing
9	NP_SEGMENT	ART N	the cytosol

Figure 2: Input representation for a sentence which contains a subcellular-localization tuple: the sentence is segmented into typed phrases and each phrase is segmented into words typed with part-of-speech tags. Phrase types and labels arc shown in column (a). Word part-of-speech tags and labels arc shown in column (b). The words of the sentence are shown in column (c). Note the grouping of words in phrases. The labels (PROTEIN, LOCATION) are present only in the training sentences.

subcellular-localization relation and its annotated segments. The sentence is segmented into typed phrases and each phrase is segmented into words typed with part-of-speech tags. For example, the second phrase segment is a noun phrase (NP.SEGMENT) that contains the protein name UBC6 (hence the PROTEIN label). Note that the types are constants that are pre-defined by our representation of Sundance parses, whereas the labels are defined by the domains of the particular relation we are trying to extract.

3 Hierarchical HMMs for Information Extraction

A schematic of one of our hierarchical HMMs is shown in Figure 3. The top of the figure shows the *positive model,* which is trained to represent sentences that contain instances of the target relation. The bottom of the figure shows the *null model,* which is trained to represent sentences that do not contain relation instances (e.g. off-topic sentences). At the "coarse" level, our hierarchical HMMs represent sentences as sequences of phrases. Thus, we can think of the top level as an HMM whose states emit phrases. We refer to this HMM as the *phrase HMM,* and its states *phrase states.* At the "fine" level, each phrase is represented as a sequence of words. This is achieved by embedding an HMM within each phrase state. We refer to these embedded HMMs as *word HMMs* and their states as *word states.* The phrase states in Figure 3 are depicted with rounded rectangles and word states are depicted with ovals. To explain a sentence, the HMM would first follow a transition from the START state to some phrase state *qi,* use the word HMM of *qi* to emit the first phrase of the sentence, then transition to another phrase state *qj,* emit another

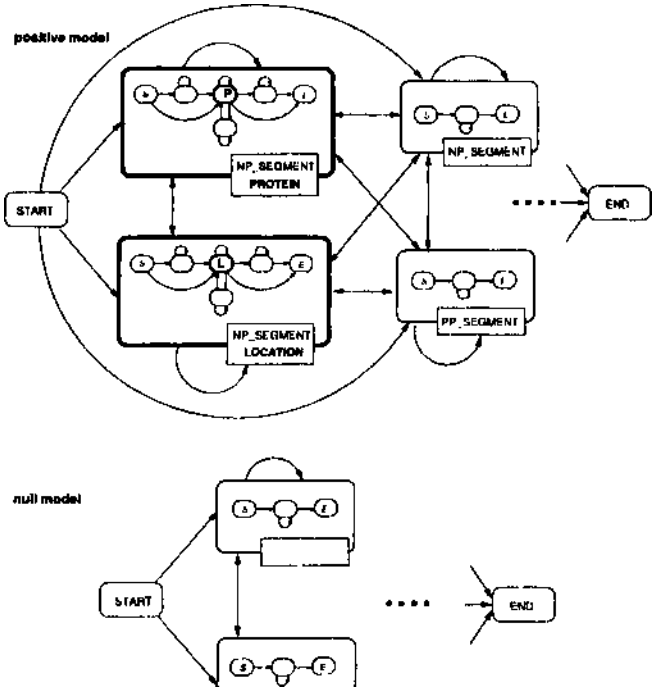


Figure 3: Schematic of the architecture of a hierarchical HMM for the subcellular-localization relation. The top part of the figure shows the positive model and the bottom part the null model. Phrase states are depicted as rounded rectangles and word states as ovals. The types and labels of the phrase states are shown within rectangles at the bottom right of each state. Labels are shown in bold and states associated with non-empty label sets are depicted with bold borders. The labels of word states are abbreviated for compactness.

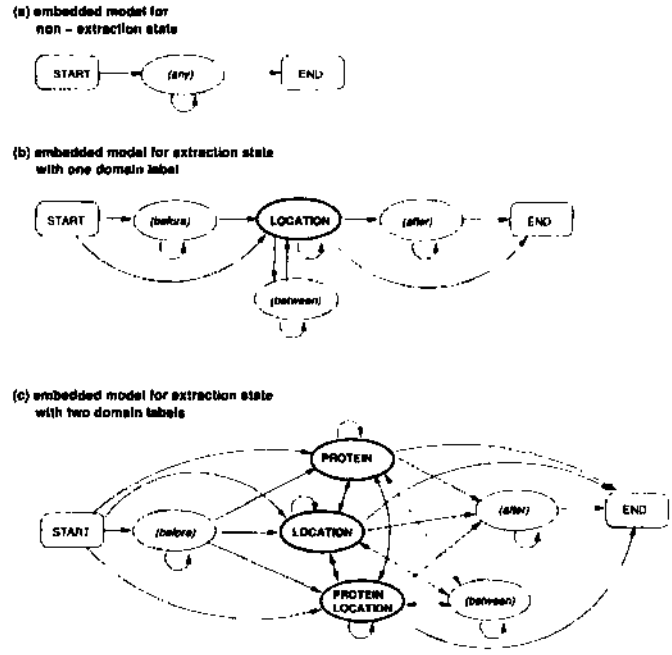


Figure 4: Architectures of the word HMMs for the subcellular-localization relation. Bold text within states denotes domain labels. For states with implicit empty labels, italicized text within parentheses denotes the position of the state's emissions relative to the domain words. The figure shows (a) the structure of the embedded HMMs for phrase states without labels, (b), phrase states with one label and (c) phrase states with two labels.

phrase using the word HMM of q_3 and so on until it moves to the END state of the phrase HMM. Note that only the word states have direct emissions.

Like the phrases in our input representation, each phrase state in the HMM has a type and may have one or more labels. Each phrase state is constrained to emit only phrases whose type agrees with the state's type. We refer to states that have labels associated with them as *extraction states,* since they are used to predict which test sentences should have tuples extracted from them.

The architectures of the word HMMs are shown in Figure 4. We use three different architectures depending on the labels associated with the phrase state in which the word HMM is embedded. The word HMMs for the phrase states with empty label sets (Figure 4(a)) consist of a single emitting state with a self-transition. For the extraction states of the phrase HMM, the word HMMs have a specialized architecture with different states for the domain instances, and for the words that come before, between and after the domain instances (Figures 4(b) and 4(c)). All the states of the word HMMs can emit words of any type (part-of-speech). That is, they are *untyped,* in contrast to the *typed* phrase states. The word states are annotated with label sets, and are trained to emit words with identical label sets. For example, the word HMM shown in Figure 4(b) can explain the phrase *"the endoplasmic reticulum "* by following a transition from the START state to the *(before)* state, emitting the word *"the",* transitioning to the LOCATION state, emitting the words *"endoplasmic"* and *"reticulum"* with the LOCATION label and then transitioning to the END state. In order for a phrase state to emit a whole phrase, as given by the input representation, and not sequences of words that are shorter or longer than a phrase, we require that the embedded word HMM transition to the end state exactly when it has emitted all the words of a given phrase. Thus word HMMs will always emit sequences of words that constitute whole phrases and transitions between phrase states occur only at phrase boundaries.

The standard dynamic programming algorithms that are used for learning and inference in HMMs - Forward, Backward and Viterbi [Rabiner, 1989] - need to be slightly modified for our hierarchical HMMs. In particular, they need to *(i)* handle the multiple-levels of the input representation, enforcing the constraint that word HMMs must emit sequences of words that constitute phrases, and *(ii)* support the use of typed phrase states by enforcing agreement between state and phrase types.

The Forward algorithm for our hierarchical HMMs is defined by the recurrence relationships shown in Table 1. The first three equations of the recurrence relation provide a phrase-level description of the algorithm, and the last three equations provide a word-level description. Notice that the third equation describes the linkage between the phrase level

$$\alpha_0(0) = 1$$
$$\alpha_a(0) = 0, \quad q_a \neq q_0$$
$$\alpha_a(i) = \begin{cases} \alpha_{a,n}(i, |s_i|), & \text{if } type(q_a) = type(s_i) \\ 0, & \text{otherwise.} \end{cases}$$
$$\alpha_{a,0}(i, 0) = \sum_b T(q_a|q_b)\alpha_b(i-1)$$
$$\alpha_{a,b}(i, 0) = 0, \quad q_{a,b} \neq q_{a,0}$$
$$\alpha_{a,b}(i, j) = E(s_{i,j}|q_{a,b}) \sum_c T(q_{a,b}|q_{a,c})\alpha_{a,c}(i, j-1)$$

s_i	i-th phrase in sentence s
$s_{i,j}$	j-th word in phrase s_i
q_a, q_0, q_n	a-th, START and END states of the phrase HMM
$q_{a,b}, q_{a,0}, q_{a,n}$	b-th, START and END states of the word HMM in phrase state q_a
$\alpha_a(i)$	probability of emitting sequence of phrases $s_1 \ldots s_i$, starting from the start state and ending at state q_a
$\alpha_{a,b}(i, j)$	probability of emitting sequence of words $s_{1,1} \ldots s_{i,j}$, starting from the START state and ending at state $q_{a,b}$
$E(s_{i,j}\|q_{a,b})$	probability that word state $q_{a,b}$ emits word $s_{i,j}$
$T(q_a\|q_b)$	probability of transition from phrase state q_b to phrase state q_a
$T(q_{a,b}\|q_{a,c})$	probability of transition from word state $q_{a,c}$ to word state $q_{a,b}$

Tabic 1: The left side of the table shows the Forward-algorithm recurrence relation for our hierarchical HMMs. The right side of the table defines the notation used in the recurrence relation.

and the word level. The Backward and Viterbi algorithms require similar modifications, but we do not show them due to space limitations.

As illustrated in Figure 2, each training instance for our HMMs consists of a sequence of words, segmented into phrases, and an associated sequence of labels. For a test instance, we would like our trained model to accurately predict a sequence of labels given only the observable part of the sentence (i.e. the words and phrases). We use a discriminative training algorithm [Krogh, 1994] that tries to find model parameters, 0, to maximize the conditional likelihood of the labels given the observable part of the sentences:

$$\hat{\theta} = \arg\max_{\theta} \prod_l \Pr(c^l|s^l, \theta). \quad (1)$$

Here s^l is the sequence of words/phrases for the Zth instance, and c^1 is the sequence of labels for the instance. This training algorithm will converge to a local maximum of the objective function. We initialize the parameters of our models by first doing standard generative training. We then apply Krogh's algorithm which involves iterative updates to the HMM parameters. To avoid overfitting, we stop training when the accuracy on a held-aside tuning set is maximized.

In order for this algorithm to be able to adjust the parameters of the positive model in response to negative instances and vice-versa, we join our positive and null models as shown in Figure 5. This combined model includes the positive and

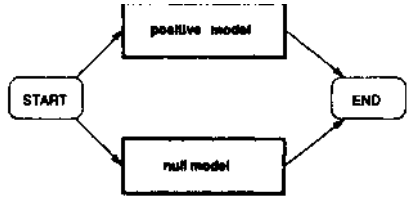


Figure 5: Architecture of the *combined model.* The positive and null models refer to the models in Figure 3.

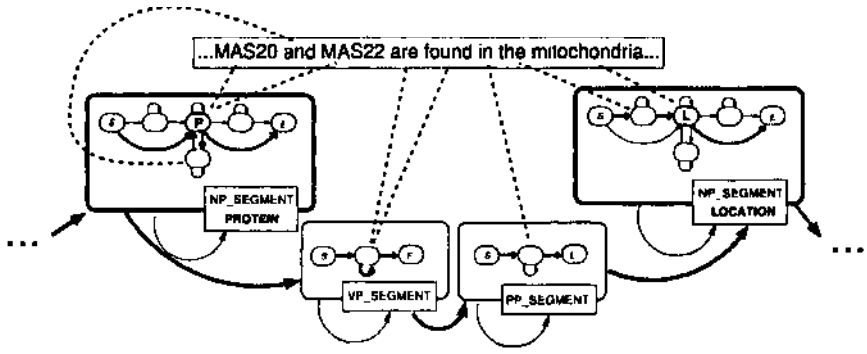
the null models (shown in Figure 3) as its two submodels, with shared START and END states.

Once a model has been trained, we can use the Viterbi algorithm to predict tuples in test sentences. We extract a tuple from a given sentence if the Viterbi path goes through states with labels for *all* the domains of the relation. For example, for the SUbcellular-localization relation, the Viterbi path for a sentence must pass through a state with the PROTEIN label and a state with the LOCATION label. This process is illustrated in Figure 6.

4 Hierarchical HMMs with Context Features

In this section we describe an extension to the hierarchical HMMs presented in the previous section that enables them to represent additional information about the structure of sentences within phrases. We refer to these extended HMMs as *Context hierarchical HMMs* (CHHMMs). Whereas the hierarchical HMMs presented earlier partition a sentence s into disjoint observations $s_{i,j}$ where each $s_{i,j}$ is a word, a CHHMM represents s as a sequence of *overlapping* observations $o_{i,j}$. Each observation $o_{i,j}$ consists of a window of three words, centered around $s_{i,j}$, together with the part-of-speech tags of these words. Formally, $o_{i,j}$ is a vector $\langle s_{i,(j-1)}, s_{i,j}, s_{i,(j+1)}, t_{i,(j-1)}, t_{i,j}, t_{i,(j+1)} \rangle$ where $t_{i,j}$ is the part-of-speech tag of word *8ij.* Note that $o_{i,j}$ and $o_{i,(j+1)}$ share $s_{i,j}, s_{i,(j+1)}, t_{i,j}$ and $t_{i,(j+1)}$, although these features are located in different positions in the two vectors. Figure 7 shows the vectors emitted for the phrase *"the endoplasmic reticulum"* by a word HMM in the CHHMM.

Using features that represent the previous and next words allows the models to capture regularities about pairs or triplets of words. For instance, a CHHMM is potentially able to learn that the word *"membrane "* is part of a subcellular location when found in *"plasma membrane"* while it is not when found in *"a membrane".* Furthermore, by using features that represent the part-of-speech of words, the models are able to learn regularities about groups of words with the same part of speech in addition to regularities about individual words.



Phrase #	Words	Word label sets	Phrase label sets
1	MAS20 and MAS22	{PROTEIN} ∅ {PROTEIN}	{PROTEIN}
2	are found	∅ ∅	∅
3	in	∅	∅
4	the mitochondria	∅ {LOCATION}	{LOCATION}

Extracted tuples: subcellular-localization(MAS20, mitochondria)
subcellular-localization(MAS22, mitochondria)

Figure 6: Example of the procedure for extracting tuples of the subcellular-localization relation from the sentence fragment "...MAS20 and MAS22 are found in the mitochondria...". The top of the figure shows how the most likely path explains the sentence fragment. Bold transitions between states denote the most likely path. Dashed lines connect each state with the words that it emits. The table shows the label sets that are assigned to the phrases and the words of the sentence. The extracted tuples arc shown at the bottom of the figure.

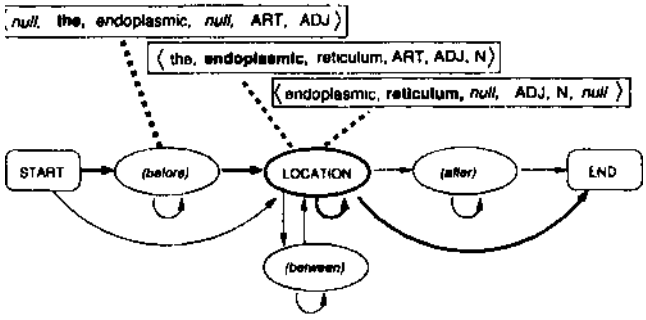


Figure 7: Generation of the phrase *"the endoplasmic reticulum"* by a word HMM in a CHHMM. The bold arcs represent the path that generates the phrase. The vector observations $o_{i,J}$ emitted by each state are shown in the rectangles above the model and are connected with dotted arcs with the emitting state. The word that would be emitted by each state of the equivalent HHMM is shown in boldface.

The advantages of this representation are especially realized when dealing with an out-of-vocabulary word; in this case part-of-speech tags and neighboring words may be quite informative about the meaning and use of the out-of-vocabulary word. For example, an out-of-vocabulary adjective will rarely be a protein, since proteins are usually nouns.

Because the number of possible observations for a given word state in a CHHMM is very large (all possible vectors representing sequences of three words and their POS tags), to model the probability of an observation $o_{i,j}$, our CHHMMs assume that the features are conditionally independent given the state. Under this assumption, the probability of the observation $o_{i,j}$ being emitted by state $q_{a,b}$ is then defined as

$$E(o_{i,j}|q_{a,b}) = \prod_{k=1..6} E_k(o_{i,j,k}|q_{a,b}) \qquad (2)$$

where $Ek(oijklqa,b)$ is the probability of word state $q_{a,b}$ emitting an observation whose A;-th feature is Oij^-

Note that the features employed by the representation of Equation 2 are clearly not conditionally independent. Consecutive words are not independent of one another and certainly the part-of-speech tag of a word is not independent of the word itself. However, we argue that the discriminative training algorithm we use [Krogh, 1994] can compensate in part for this violation of the independence assumption.

5 Empirical Evaluation

In this section we present experiments testing the hypothesis that our hierarchical HMMs are able to provide more accurate models than HMMs that incorporate less grammatical information. In particular we empirically compare two types of hierarchical HMMs with three baseline HMMs.

- Context HHMMs: hierarchical HMMs with context features, as described in the previous section.
- HHMMs: hierarchical HMMs without context features.
- Phrase HMMs: single-level HMMs in which states are typed (as in the phrase level of an HHMM) and emit whole phrases. These HMMs were introduced by Ray and Craven (2001). Unlike hierarchical HMMs, the states of Phrase HMMs do not have embedded HMMs which emit words. Instead each state has a single multinomial distribution to represent its emissions, and each emitted phrase is treated as a bag of words.

- POS HMMs: single-level HMMs in which states emit words, but are typed with part-of-speech tags so that a given state can emit words with only a single POS.
- Token HMMs: single-level HMMs in which untyped states emit words.

We evaluate our hypotheses on three data sets that we have assembled from the biomedical literature.[1] The data sets are composed of abstracts gathered from the MEDLINE database [National Library of Medicine, 2003]. The first set contains instances of the subcellular-localization relation. It is composed of 769 positive and 6,360 negative sentences. The positive sentences contain 949 total tuple instances. The number of actual tuples is 404 since some tuples occur multiple times either in the same sentence or in multiple sentences. The second, which we refer to as the disorder-association data set, characterizes a binary relation between genes and disorders. It contains 829 positive and 11,771 negative sentences. The positive sentences represent 878 instances of 145 tuples. The third, which we refer to as the protein-interaction data set, characterizes physical interactions between pairs of proteins. It is composed of 5,457 positive and 42,015 negative sentences. It contains 8,088 instances of 8i9 tuples.

We use five-fold cross-validation to measure the accuracy of each approach. Before processing all sentences, we obtain parses from Sundance, and then stem words with Porter's stemmer [Porter, 1980]. We map all numbers to a special NUMBER token and all words that occur only once in a training set to an OUT-OF-VOCAB token. Also, we discard all punctuation. The same preprocessing is done on test sentences, with the exception that words that were not encountered in the training set are mapped to the OUT-OF-VOCAB token. The vocabulary is the same for all emitting states in the models, and all parameters are smoothed using *m-estimates* [Cestnik, 1990J. We train all models using the discriminative training procedure referred to in Section 3 [Krogh, 1994].

To evaluate our models we construct *precision-recall* graphs. *Precision* is defined as the fraction of correct tuple instances among those instances that are extracted by the model. *Recall* is defined as the fraction of correct tuple instances extracted by the model over the total number of tuple instances that exist in the data set. For each tuple extracted from sentence s, we calculate a confidence measure as:

$$c(s) = \frac{\delta_n(|s|)}{\alpha_n(|s|)}.$$

Here q_n refers to the END state of the combined model, $<5_n(|s|)$ is the probability of the most likely path, given by the Viterbi algorithm, and $a_n(N)$ is the total probability of the sequence, calculated with the Forward algorithm. We construct precision-recall curves by varying a threshold on these confidences.

Figures 8, 9 and 10 show the precision-recall curves for the three data sets. Each figure shows curves for the five types of HMMs described at the beginning of this section. We show error bars for the Context HHMM precision values for the subcellular-localization and protein-interaction data sets. For these two data sets, the hierarchical HMM models clearly have superior precision-recall curves to the baseline models. At nearly every level of recall, the hierarchical HMMs exhibit higher precision than the baselines. Additionally, the HHMMs achieve higher endpoint recall values. The results are not as definitive for the disorder-association data set. Here, the POS HMMs and the Token HMMs achieve precision levels that are comparable to, and in some cases slightly better than, the Context HHMMs. There is not a clear winner for this data set, but the Context HHMMs are competitive.

Comparing the Context HHMMs to the ordinary HHMMs, we see that the former results in superior precision-recall curves for all three data sets. This result demonstrates that clearly there is value in including the context features in hierarchical HMMs for this type of task. In summary, our empirical results support the hypothesis that the ability our hierarchical HMM approach to capture grammatical information about sentences results in more accurate learned models.

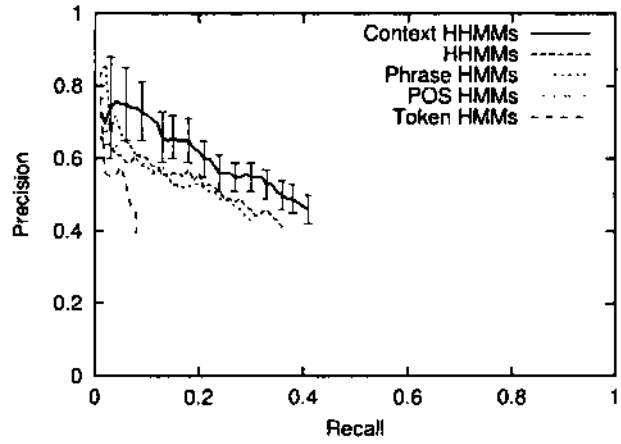


Figure 8: Precision vs. recall for the five types of HMMs on the subcellular-localization data set.

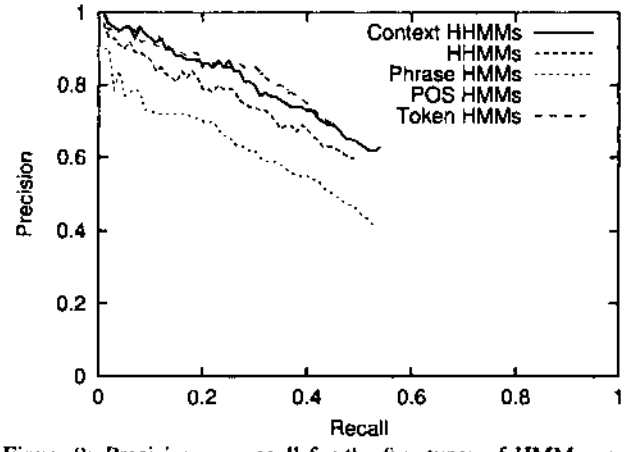


Figure 9: Precision vs. recall for the five types of HMMs on the disorder-association data set.

Earlier versions of two of these data sets were used in our previous work [Ray and Craven, 2001]. Various aspects of the data sets have been cleaned up, however, and thus the versions used here are somewhat different. All three data sets arc available from http://www.biostat.wise.edu/"craven/ie/.

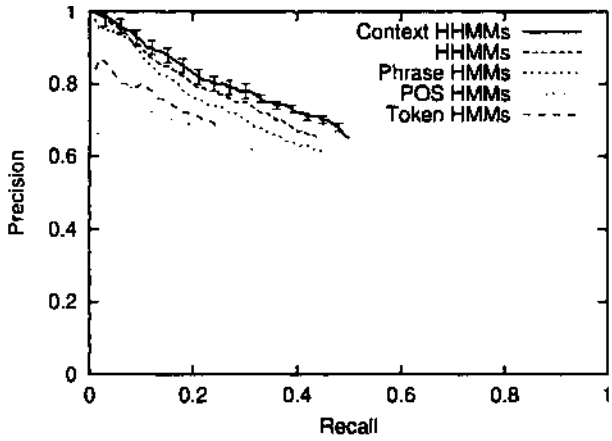


Figure 10: Precision vs. recall for the five types of HMMs on the protein-interaction data set.

6 Conclusion

We have presented an approach to learning models for information extraction that is based on using hierarchical HMMs to represent the grammatical structure of the sentences being processed. We employ a shallow parser to obtain parse trees for sentences and then use these trees to construct the input representation for the hierarchical HMMs

Our approach builds on previous work on hierarchical HMMs and incorporating grammatical knowledge into information-extraction models. The application of HHMMs to IE is novel and has required us to modify HHMM learning algorithms to operate on a hierarchical input representation. In particular our methods take into account that phrases and states must have matching types, and that phrase states must emit complete phrases. We have also introduced a novel modification of HHMMs in which observations can be feature vectors. With respect to previous work on incorporating grammatical knowledge into IE models, our main contribution is an approach that takes advantage of grammatical information represented at multiple scales. An appealing property of our approach is that it generalizes to additional levels of description of the input text.

We have evaluated our approach in the context of learning IE models to extract instances of three biomedical relations from the abstracts of scientific articles. These experiments demonstrate that incorporating a hierarchical representation of grammatical structure improves extraction accuracy in hidden Markov models.

Acknowledgments

This research was supported in part by NIH grant 1R01 LM07050-01, NSF grant IIS-0093016, and a grant to the University of Wisconsin Medical School under the Howard Hughes Medical Institute Research Resources Program for Medical Schools.

References

[Bikelrfa/., 1999] D. Bikel, R. Schwartz, and R. Weischedel. An algorithm that learns what's in a name. *Machine Learning,* 34(1):211-231, 1999.

fCestnik, 1990] B. Cestnik. Estimating probabilities: A crucial task in machine learning. In *Proc. of the 9th European Conf on Artificial Intelligence,* pages 147-150, Stockholm, Sweden, 1990. Pitman.

[Fineet al., 1998] S. Fine, Y. Singer, and N. Tishby. The hierarchical hidden Markov model: Analysis and applications. *Machine Learning,* 32:41-62, 1998.

[Freitag and McCallum, 2000] D. Freitag and A. McCallum. Information extraction with HMM structures learned by stochastic optimization. In *Proc. of the 17th National Conf on Artificial Intelligence,* 2000. AAAI Press.

[Hirschman *etal,* 2002] L. Hirschman, J. Park, J. Tsujii, L. Wong, and C. Wu. Accomplishments and challenges in literature data mining for biology. *Bioinformatics,* 18:1553-1561,2002.

[Krogh, 1994] A. Krogh. Hidden Markov models for labeled sequences. In *Proc. of the 12th International Conf on Pattern Recognition,* pages 140-144, Jerusalem, Israel, 1994. IEEE Computer Society Press.

[Laffertyet al., 2001] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of the 18th International Conf on Machine Learning,* pages 282-289, Williamstown, MA, 2001. Morgan Kaufmann.

[Leek, 1997] T. Leek. Information extraction using hidden Markov models. M.S. thesis, Dept. of Computer Science and Engineering, Univ. of California, San Diego, 1997.

lMcCallum et al.,2000] A. McCallum, D. Freitag, and F. Pereira. Maximum entropy Markov models for information extraction and segmentation. In *Proc. of the 17th International Conf on Machine Learning,* pages 591-598, Stanford, CA, 2000. Morgan Kaufmann.

[Miller *etal,* 2000] S. Miller, H. Fox, L. Ramshaw, and R. Weischedel. A novel use of statistical parsing to extract information from text. In *Proc. of the 6th Applied Natural Language Processing Conf,* pages 226-233, Seattle, WA, 2000. Association for Computational Linguistics.

[National Library of Medicine, 2003] National Library of Medicine. The MEDLINE database, 2003. http://www.ncbi.nlm.nih.gov/PubMed/.

[Porter, 1980] M. F. Porter. An algorithm for suffix stripping. *Program,* 14(3): 127-130, 1980.

[Rabiner, 1989] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. of the IEEE,* 77(2):257-286,1989.

[Ray and Craven, 2001] S. Ray and M. Craven. Representing sentence structure in hidden Markov models for information extraction. In *Proc. of the 17th IntemationalJoint Conf on Artificial Intelligence,* pages 1273-1279, Seattle, WA, 2001. Morgan Kaufmann.

[Riloff, 1998] E. Riloff. The sundance sentence analyzer, 1998. http://www.cs.utah.edu/projects/nlp/.