Jointly Labeling Multiple Sequences: A Factorial HMM Approach

Kevin Duh
Department of Electrical Engineering
University of Washington, USA
duh@ee.washington.edu

Abstract

We present new statistical models for jointly labeling multiple sequences and apply them to the combined task of part-of-speech tagging and noun phrase chunking. The model is based on the Factorial Hidden Markov Model (FHMM) with distributed hidden states representing part-of-speech and noun phrase sequences. We demonstrate that this joint labeling approach, by enabling information sharing between tagging/chunking subtasks, outperforms the traditional method of tagging and chunking in succession. Further, we extend this into a novel model, Switching FHMM, to allow for explicit modeling of cross-sequence dependencies based on linguistic knowledge. We report tagging/chunking accuracies for varying dataset sizes and show that our approach is relatively robust to data sparsity.

1 Introduction

Traditionally, various sequence labeling problems in natural language processing are solved by the cascading of well-defined subtasks, each extracting specific knowledge. For instance, the problem of information extraction from sentences may be broken into several stages: First, part-of-speech (POS) tagging is performed on the sequence of word tokens. This result is then utilized in noun-phrase and verb-phrase chunking. Finally, a higher-level analyzer extracts relevant information based on knowledge gleaned in previous subtasks.

The decomposition of problems into well-defined subtasks is useful but sometimes leads to unnecessary errors. The problem is that errors in earlier subtasks will propagate to downstream subtasks, ultimately deteriorating overall performance. Therefore, a method that allows the *joint labeling* of subtasks is desired. Two major advantages arise from simultaneous labeling: First, there is more robustness against error propagation. This is especially relevant if we use probabilities in our models. Cascading subtasks inherently "throws away" the probability at each stage; joint labeling preserves the uncertainty. Second, information between simultaneous subtasks can be shared to further improve accuracy. For instance, it is possible that knowing a certain noun phrase chunk may help the model infer POS tags more accurately, and vice versa.

In this paper, we propose a solution to the joint labeling problem by representing multiple sequences in a single Factorial Hidden Markov Model (FHMM) (Ghahramani and Jordan, 1997). The FHMM generalizes hidden Markov models (HMM) by allowing separate hidden state sequences. In our case, these hidden state sequences represent the POS tags and phrase chunk labels. The links between the two hidden sequences model dependencies between tags and chunks. Together the hidden sequences generate an observed word sequence, and the task of the tagger/chunker is to invert this process and infer the original tags and chunks.

Previous work on joint tagging/chunking has shown promising results. For example, Xun et

Proceedings of the ACL Student Research Workshop, pages 19–24,
Ann Arbor, Michigan, June 2005.

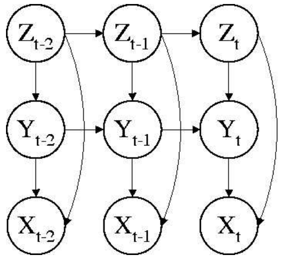


Figure 1: Baseline FHMM. The two hidden sequences $y_{1:t}$ and $z_{1:t}$ can represent tags and chunks, respectively. Together they generate $x_{1:t}$, the observed word sequence.

al. (2000) uses a POS tagger to output an N-best list of tags, then a Viterbi search to find the chunk sequence that maximizes the joint tag/chunk probability. Florian and Ngai (2001) extends transformation-based learning tagger to a joint tagger/chunker by modifying the objective function such that a transformation rule is evaluated on the classification of all simultaneous subtasks. Our work is most similar in spirit to Dynamic Conditional Random Fields (DCRF) (Sutton et al., 2004), which also models tagging and chunking in a factorial framework. Some main differences between our model and DCRF may be described as 1) directed graphical model vs. undirected graphical model, and 2) generative model vs. conditional model. The main advantage of FHMM over DCRF is that FHMM requires considerably less computation and exact inference is easily achievable for FHMM and its variants.

The paper is structured as follows: Section 2 describes in detail the FHMM. Section 3 presents a new model, the Switching FHMM, which represents cross-sequence dependencies more effectively than FHMMs. Section 4 discusses the task and data and Section 5 presents various experimental results Section 6 discusses future work and concludes.

2 Factorial HMM

2.1 Basic Factorial HMM

A Factorial Hidden Markov Model (FHMM) is a hidden Markov model with a distributed state representation. Let $x_{1:T}$ be a length T sequence of observed random variables (e.g. words) and $y_{1:T}$ and $z_{1:T}$ be the corresponding sequences of hidden state variables (e.g. tags, chunks). Then we define the FHMM as the probabilistic model:

$$\begin{aligned} & p(x_{1:T}, y_{1:T}, z_{1:T}) \\ = \quad & \pi_0 \prod_{t=2}^{T} p(x_t|y_t, z_t) p(y_t|y_{t-1}, z_t) p(z_t|z_{t-1}) \end{aligned} \tag{1}$$

where $\pi_0 = p(x_0|y_0, z_0)p(y_0|z_0)p(z_0)$. Viewed as a generative process, we can say that the **chunk model** $p(z_t|z_{t-1})$ generates chunks depending on the previous chunk label, the **tag model** $p(y_t|y_{t-1}, z_t)$ generates tags based on the previous tag and current chunk, and the **word model** $p(x_t|y_t, z_t)$ generates words using the tag and chunk at the same time-step.

This equation corresponds to the graphical model of Figure 1. Although the original FHMM developed by Ghahramani (1997) does not explicitly model the dependencies between the two hidden state sequences, here we add the edges between the y and z nodes to reflect the interaction between tag and chunk sequences. Note that the FHMM can be collapsed into a hidden Markov model where the hidden state is the cross-product of the distributed states y and z. Despite this equivalence, the FHMM is advantageous because it requires the estimation of substantiatially fewer parameters.

FHMM parameters can be calculated via maximum likelihood (ML) estimation if the values of the hidden states are available in the training data. Otherwise, parameters must be learned using approximate inference algorithms (e.g. Gibbs sampling, variational inference), since exact Expectation-Maximization (EM) algorithm is computationally intractable (Ghahramani and Jordan, 1997). Given a test sentence, inference of the corresponding tag/chunk sequence is found by the Viterbi algorithm, which finds the tag/chunk sequence that maximizes the joint probability, i.e.

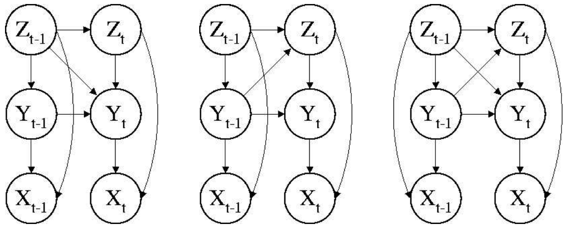
$$\arg \max_{y_{1:T}, z_{1:T}} p(x_{1:T}, y_{1:T}, z_{1:T}) \tag{2}$$

2.2 Adding Cross-Sequence Dependencies

Many other structures exist in the FHMM framework. Statistical modeling often involves the iterative process of finding the best set of dependencies that characterizes the data effectively. As shown in Figures 2(a), 2(b), and 2(c), dependencies can be added between the y_t and z_{t-1}, between z_t and y_{t-1}, or both. The model in Fig. 2(a) corresponds to changing the tag model in Eq. 1 to $p(y_t|y_{t-1}, z_t, z_{t-1})$; Fig. 2(b) corresponds to changing the chunk model to $p(z_t|z_{t-1}, y_{t-1})$; Fig. 2(c), corresponds to changing both tag and chunk models, leading to the probability model:

$$\prod_{t=1}^{T} p(x_t|y_t, z_t)p(y_t|y_{t-1}, z_t, z_{t-1})p(z_t|z_{t-1}, y_{t-1}) \tag{3}$$

We name the models in Figs. 2(a) and 2(b) as FHMM-T and FHMM-C due to the added dependencies to the **t**ag and **c**hunk models, respectively. The model of Fig. 2(c) and Eq. 3 will be referred to as FHMM-CT. Intuitively, the added dependencies will improve the predictive power across chunk and tag sequences, provided that enough training data are available for robust parameter estimation.



(a) (b) (c)

Figure 2: FHMMs with additional cross-sequence dependencies. The models will be referred to as (a) FHMM-T, (b) FHMM-C, and (c) FHMM-CT.

3 Switching Factorial HMM

A reasonable question to ask is, "How exactly does the chunk sequence interact with the tag sequence?" The approach of adding dependencies in Section 2.2 acknowledges the existence of cross-sequence interactions but does not explicitly specify the type of interaction. It relies on statistical learning to find the salient dependencies, but such an approach is feasable only when sufficient data are available for parameter estimation.

To answer the question, we consider how the chunk sequence affects the generative process for tags: First, we can expect that the unigram distribution of tags changes depending on whether the chunk is a noun phrase or verb phrase. (In a noun phrase, nouns and adjective tags are more common; in a verb phrase, verbs and adverb tags are more frequent.) Similarly, a bigram distribution $p(y_t|y_{t-1})$ describing tag transition probabilities differs depending on the bigram's location in the chunk sequence, such as whether it is within a noun phrase, verb phrase, or at a phrase boundary. In other words, the chunk sequence interacts with tags by switching the particular generative process for tags. We model this interaction explicitly using a Switching FHMM:

$$\begin{aligned} & p(x_{1:T}, y_{1:T}, z_{1:T}) \\ &= \prod_{t=1}^{T} p(x_t|y_t, z_t)p_\alpha(y_t|y_{t-1})p_\beta(z_t|z_{t-1}) \end{aligned} \tag{4}$$

In this new model, the chunk and tag are now generated by bigram distributions parameterized by α and β. For different values of α (or β), we have different distributions for $p(y_t|y_{t-1})$ (or $p(z_t|z_{t-1})$). The crucial aspect of the model lies in a function $\alpha = f(z_{1:t})$, which summarizes information in $z_{1:t}$ that is relevant for the generation of y, and a function $\beta = g(y_{1:t})$, which captures information in $y_{1:t}$ that is relevant to the generation of z.

In general, the functions $f(\cdot)$ and $g(\cdot)$ partition the space of all tag or chunk sequences into several equivalence classes, such that all instances of an equivalence class give rise to the same generative model for the cross sequence. For instance, all consecutive chunk labels that indicate a noun phrase can be mapped to one equivalence class, while labels that indicate verb phrase can be mapped to another. The mapping can be specified manually or learned automatically. Section 5 discusses a linguistically-motivated mapping that is used for the experiments.

Once the mappings are defined, the parameters $p_\alpha(y_t|y_{t-1})$ and $p_\beta(z_t|z_{t-1})$ are obtained via maximum likelihood estimation in a fashion similar to that of the FHMM. The only exception is that now the training data are partitioned according to the mappings, and each α- and β- specific generative model is estimated separately. Inference of the tags and chunks for a test sentence proceeds similarly to FHMM inference. We call this model a Switching FHMM since the distribution of a hidden sequence "switches" dynamically depending on the values of the other hidden sequence.

An idea related to the Switching FHMM is the Bayesian Multinet (Geiger and Heckerman, 1996;

Bilmes, 2000), which allows the dynamic switching of conditional variables. It can be used to implement switching from a higher-order model to a lower-order model, a form of backoff smoothing for dealing with data sparsity. The Switching FHMM differs in that it switches among models of the same order, but these models represent different generative processes. The result is that the model no longer requires a time-homogenous assumption for state transitions; rather, the transition probabilities change dynamically depending on the influence across sequences.

4 POS Tagging and NP Chunking

4.1 The Tasks

POS tagging is the task of assigning words the correct part-of-speech, and is often the first stage of various natural language processing tasks. As a result, POS tagging has been one of the most active areas of research, and many statistical and rule-based approach have been tried. The most notable of these include the trigram HMM tagger (Brants, 2000), maximum entropy tagger (Ratnaparkhi, 1996), transformation-based tagger (Brill, 1995), and cyclic dependency networks (Toutanova et al., 2003).

Accuracy numbers for POS tagging are often reported in the range of 95% to 97%. Although this may seem high, note that a tagger with 97% accuracy has only a 63% chance of getting all tags in a 15-word sentence correct, whereas a 98% accurate tagger has 74% (Manning and Schütze, 1999). Therefore, small improvements can be significant, especially if downstream processing requires correctly-tagged sentences. One of the most difficult problems with POS tagging is the handling of out-of-vocabulary words.

Noun-phrase (NP) chunking is the task of finding the non-recursive (base) noun-phrases of sentences. This segmentation task can be achieved by assigning words in a sentence to one of three tokens: B for "Begin-NP", I for "Inside-NP", or O for "Outside-NP" (Ramshaw and Marcus, 1995). The "Begin-NP" token is used in the case when an NP chunk is immediately followed by another NP chunk. The state-of-the-art chunkers report F1 scores of 93%-94% and accuracies of 87%-97%. See, for example, NP chunkers utilizing conditional random fields (Sha and Pereira, 2003) and support vector machines (Kudo and Matsumoto, 2001).

4.2 Data

The data comes from the CoNLL 2000 shared task (Sang and Buchholz, 2000), which consists of sentences from the Penn Treebank Wall Street Journal corpus (Marcus et al., 1993). The training set contains a total of 8936 sentences with 19k unique vocabulary. The test set contains 2012 sentences and 8k vocabulary. The out-of-vocabulary rate is 7%.

There are 45 different POS tags and 3 different NP labels in the original data. An example sentence with POS and NP tags is shown in Table 1.

The	move	could	pose	a	challenge
DT	NN	MD	VB	DT	NN
I	I	O	O	I	I

Table 1: Example sentence with POS tags (2nd row) and NP labels (3rd row). For NP, I = Inside-NP, O=Outside-NP.

5 Experiments

We report two sets of experiments. Experiment 1 compares several FHMMs with cascaded HMMs and demonstrates the benefit of joint labeling. Experiment 2 evaluates the Switching FHMM for various training dataset sizes and shows its robustness against data sparsity. All models are implemented using the Graphical Models Toolkit (GMTK) (Bilmes and Zweig, 2002).

5.1 Exp1: FHMM vs Cascaded HMMs

We compare the four FHMMs of Section 2 to the traditional approach of cascading HMMs in succession, and compare their POS and NP accuracies in Table 2. In this table, the first row "Oracle HMM" is an oracle experiment which shows what NP accuracies can be achieved if perfectly correct POS tags are available in a cascaded approach. The second row "Cascaded HMM" represents the traditional approach of doing POS tagging and NP chunking in succession; i.e. an NP chunker is applied to the output of a POS tagger that is 94.17% accurate. The next four rows show the results of joint labeling using various FHMMs. The final row "DCRF" are comparable results from Dynamic Conditional Random Fields (Sutton et al., 2004).

There are several observations: First, it is important to note that FHMM outperforms the cascaded HMM in terms of NP accuracy for all but one model. For instance, FHMM-CT achieves an NP accuracy of 95.93%, significantly higher than both the cascaded HMM (93.90%) and the oracle HMM (94.67%). This confirms our hypothesis that joint labeling helps prevent POS errors from propagating to NP chunking. Second, the fact that several FHMM models achieve NP accuracies higher than the *oracle* HMM implies that information sharing between POS and NP sequences gives even more benefit than having only perfectly correct POS tags. Thirdly, the fact that the most complex model (FHMM-CT) performs best suggests that it is important to avoid data sparsity problems, as it requires more parameters to be estimated in training.

Finally, it should be noted that although the DCRF outperforms the FHMM in this experiment, the DCRF uses significantly more word features (e.g. capitalization, existence in a list of proper nouns, etc.) and a larger context (previous and next 3 tags), whereas the FHMM considers the word as its sole feature, and the previous tag as its only context. Further work is required to see whether the addition of these features in the FHMM's generative framework will achieve accuracies close to that of DCRF. The take-home message is that, in light of the computational advantages of generative models, the FHMM should not be dismissed as a potential solution for joint labeling. In fact, recent results in the discriminative training of FHMMs (Bach and Jordan, 2005) has shown promising results in speech processing and it is likely that such advanced techniques, among others, may improve the FHMM's performance to state-of-the-art results.

5.2 Exp2: Switching FHMM and Data Sparsity

We now compare the Switching FHMM to the best model of Experiment 1 (FHMM-CT) for varying amounts of training data. The Switching FHMM uses the following α and β mapping. The mapping $\alpha = f(z_{1:t})$ partitions the space of chunk history $z_{1:t}$ into five equivalence classes based on the two most recent chunk labels:

Model	POS	NP
Oracle HMM	–	94.67
Cascaded HMM	94.17	93.90
Baseline FHMM	93.82	93.56
FHMM-T	93.73	94.07
FHMM-C	**94.16**	95.76
FHMM-CT	94.15	**95.93**
DCRF	98.92	97.36

Table 2: POS and NP Accuracy for Cascaded HMM and FHMM Models.

Class1. $\{z_{1:t} : z_{t-1} = I, z_t = I\}$
Class2. $\{z_{1:t} : z_{t-1} = O, z_t = O\}$
Class3. $\{z_{1:t} : z_{t-1} = \{I, B\}, z_t = O\}$
Class4. $\{z_{1:t} : z_{t-1} = O, z_t = \{I, B\}\}$
Class5. $\{z_{1:t} : (z_{t-1}, z_t) = \{(I, B), (B, I)\}\}$

Class1 and Class2 are cases where the tag is located strictly inside or outside an NP chunk. Class3 and Class4 are situations where the tag is leaving or entering an NP, and Class5 is when the tag transits between consecutive NP chunks. Class-specific tag bigrams $p_\alpha(y_t|y_{t-1})$ are trained by dividing the training data according to the mapping. On the other hand, the mapping $\beta = g(y_{1:t})$ is not used to ensure a single point of comparison with FHMM-CT; we use FHMM-CT's chunk model $p_(z_t|z_{t-1}, y_{t-1})$ in place of $p_\beta(z_t|z_{t-1})$.

The POS and NP accuracies are plotted in Figures 3 and 4. We report accuracies based on the average of five different random subsets of the training data for datasets of sizes 1000, 3000, 5000, and 7000 sentences. Note that for the Switching FHMM, POS and NP accuracy remains relatively constant despite the reduction in data size. This suggests that a more explicit model for cross sequence interaction is essential especially in the case of insufficient training data. Also, for the very small datasize of 1000, the accuracies for Cascaded HMM are 84% for POS and 70% for NP, suggesting that the general FHMM framework is still beneficial.

6 Conclusion and Future Work

We have demonstrated that joint labeling with an FHMM can outperform the traditional approach of cascading tagging and chunking in NLP. The new Switching FHMM generalizes the FHMM by allow-

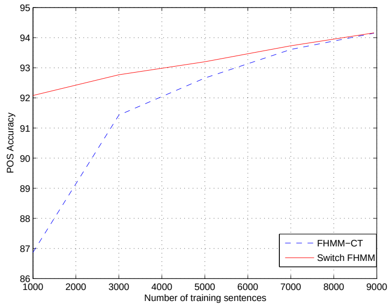


Figure 3: POS Accuracy for varying data sizes

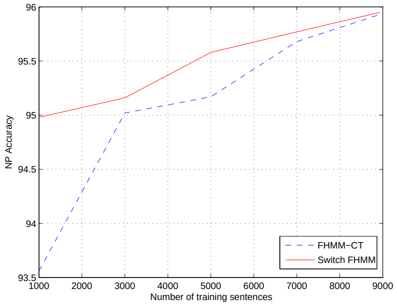


Figure 4: NP Accuracy for varying data sizes

ing dynamically changing generative models and is a promising approach for modeling the type of interactions between hidden state sequences.

Three directions for future research are planned: First, we will augment the FHMM such that its accuracies are competitive with state-of-the-art taggers and chunkers. This includes adding word features to improve accuracy on OOV words, augmenting the context from bigram to trigram, and applying advanced smoothing techniques. Second, we plan to examine the Switching FHMM further, especially in terms of automatic construction of the α and β function. A promising approach is to learn the mappings using decision trees or random forests, which has recently achieved good results in a similar problem in language modeling (Xu and Jelinek, 2004). Finally, we plan to integrate the tagger/chunker in an end-to-end system, such as a Factored Language Model (Bilmes and Kirchhoff, 2003), to measure the overall merit of joint labeling.

Acknowledgments

The author would like to thank Katrin Kirchhoff, Jeff Bilmes, and Gang Ji for insightful discussions, Chris Bartels for support on GMTK, and the two anonymous reviewers for their constructive comments. Also, the author gratefully acknowledges support from NSF and CIA under NSF Grant No. IIS-0326276.

References

Francis Bach and Michael Jordan. 2005. Discriminative training of hidden Markov models for multiple pitch tracking. In *Proc. Intl. Conf. Acoustics, Speech, Signal Processing*.

J. Bilmes and K. Kirchhoff. 2003. Factored language models and generalized parallel backoff. In *Proc. of HLT/NACCL*.

J. Bilmes and G. Zweig. 2002. The Graphical Models Toolkit: An open source software system for speech and time-series processing. In *Intl. Conf. on Acoustics, Speech, Signal Proc.*

Jeff Bilmes. 2000. Dynamic bayesian multi-networks. In *The 16th Conference on Uncertainty in Artificial Intelligence*.

Thorsten Brants. 2000. TnT – a statistical part-of-speech tagger. In *Proceedings of the Applied NLP*.

Eric Brill. 1995. Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. *Computational Linguistics*, 21(4):543–565.

Radu Florian and Grace Ngai. 2001. Multidimensional transformation-based learning. In *Proc. CoNLL*.

D. Geiger and D. Heckerman. 1996. Knowledge representation and inference in similarity netwrosk and Bayesian multinets. *Artificial Intelligence*, 82:45–74.

Z. Ghahramani and M. I. Jordan. 1997. Factorial hidden Markov models. *Machine Learning*, 29:245–275.

T. Kudo and Y. Matsumoto. 2001. Chunking with support vector machines. In *Proceedings of NAACL-2001*.

C. D. Manning and H. Schütze, 1999. *Foundations of Statistical Natural Language Processing*, chapter 10. MIT Press.

M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19:313–330.

L. A. Ramshaw and M. P. Marcus. 1995. Text chunking using transformation-based learning. In *Proceedings of the Third Workshop on Very Large Corpora (ACL-95)*.

A. Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In *Proceedings of EMNLP-1996*.

E. F. Tjong Kim Sang and S. Buchholz. 2000. Introduction to the CoNLL-2000 shared task: Chunking. In *Proc. CoNLL*.

Fei Sha and Fernando Pereira. 2003. Shallow parsing with conditional random fields. In *Proceedings of HLT-NAACL*.

C. Sutton, K. Rohanimanesh, and A. McCallum. 2004. Dynamic conditional random fields. In *Intl. Conf. Machine Learning (ICML 2004)*.

K. Toutanova, D. Klein, C. Manning, and Y. Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proc. of HLT-NAACL*.

Peng Xu and Frederick Jelinek. 2004. Random forests in language modeling. In *Proc. EMNLP*.

E. Xun, C. Huang, and M. Zhou. 2000. A unified statistical model for the identification of English BaseNP. In *Proc. ACL*.