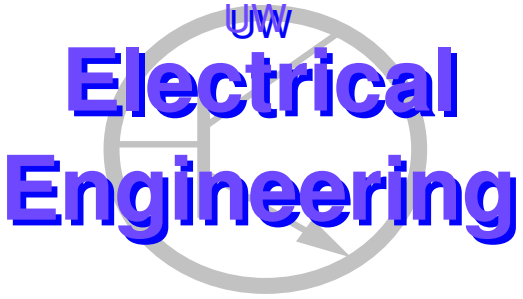
Backoff Model Training using Partially Observed Data: Application to Dialog Act Tagging

Gang Ji, Jeff Bilmes
{gang,bilmes}@ee.washington.edu

Dept of EE, University of Washington
Seattle WA, 98195-2500



UWEE Technical Report
Number UWEETR-2005-0008
August 2005

Department of Electrical Engineering
University of Washington
Box 352500
Seattle, Washington 98195-2500
PHN: (206) 543-2150
FAX: (206) 543-3842
URL: http://www.ee.washington.edu

Backoff Model Training using Partially Observed Data: Application to Dialog Act Tagging

Gang Ji, Jeff Bilmes
{gang,bilmes}@ee.washington.edu

Dept of EE, University of Washington
Seattle WA, 98195-2500

University of Washington, Dept. of EE, UWEETR-2005-0008

August 2005

Dialog act (DA) tags are useful for many applications in natural language processing and automatic speech recognition. In this work, we introduce *hidden backoff models* (HBMs) where a large generalized backoff model is trained, using an embedded expectation-maximization (EM) procedure, on data that is only partially observed. We use HBMs as word models conditioned on both DAs and (hidden) DA-segments. Experimental results on the ICSI meeting recorder dialog act (MRDA) corpus show that our embedded EM algorithm can strictly increase log likelihood on training data and can effectively reduce the error rate on test data. Different improvements are shown using different numbers of hidden states for each DA. In the best case, test error can be reduced by 6.1% relative to our baseline, and is competitive with other models even without using acoustic prosody.

1 Introduction

Discourse patterns in natural conversations and meetings are well known to provide interesting and useful information about human conversational behavior. They thus attract research from many different and beneficial perspectives. Dialog acts (DAs) [22], which reflect the functions that utterances serve in a discourse, are one type of such patterns. Detecting and understanding dialog act patterns can provide benefit to systems such as automatic speech recognition (ASR) [27], machine dialog translation [12], and general natural language processing (NLP) [10, 8]. Dialog act pattern recognition is an instance of "tagging." Many different techniques have been quite successful in this endeavor, including hidden Markov models [9, 27], semantic classification trees and polygrams [15], maximum entropy models [1], and other language models [20, 21]. Like other tagging tasks, dialog act recognition can also be achieved using conditional random fields [11, 28] and general discriminative modeling on structured outputs [2]. In many sequential data analysis tasks (speech, language, or DNA sequence analysis), standard dynamic Bayesian networks (DBNs) [17] have shown great flexibility and are widely used.

Most DA classification procedures assume that within a sentence of a particular fixed DA type, there is a fixed word distribution over the entire sentence. Similar to [14] (and see citations therein), we have found, however, that intra-sentence discourse patterns are inherently dynamic. Moreover, the patterns are specific to each type of DA, meaning a sentence will go through a DA-specific sequence of sub-DA phases or "states." A generative description of this phenomena is that a DA is first chosen, and then words are generated according to both the DA and to the relative position of the word in that sentence. For example, a

"statement" (one type of DA) can consist of a subject (noun phrase), verb phrase, and object (noun phrase). This particular sequence might be different for a different DA (e.g., a "back-channel"). Our belief is that explicitly modeling these internal states can help a DA-classification system in conversational meetings or dialogs.

In this work, we describe an approach that is motivated by several aspects of the typical DA-classification procedure. First, it is rare to have sub-DAs labeled in training data, and indeed this is true of the corpus [24] that we use. Therefore, some form of unsupervised clustering or pre-shallow-parsing of sub-DAs must be performed. In such a model, these sub-DAs are essentially unknown hidden variables that ideally could be trained with an expectation-maximization (EM) procedure. Second, when training models of language, it is necessary to employ some form of smoothing methodology since otherwise data-sparseness would render standard maximum-likelihood trained models useless. Third, discrete conditional probability distributions formed using backoff models that have been smoothed (particularly using modified Kneser-Ney [5]) have been extremely successful in many language modeling tasks. Training backoff models, however, requires that all data is observed so that data counts can be formed. Indeed, our DA-specific word models (implemented via backoff) will also need to condition on the current sub-DA, which at training time is unknown. We therefore have developed a procedure that allows us to train generalized backoff models even when some or all of the variables involved in the model are *hidden*. We thus call our models *hidden backoff model*s (HBMs). Our method is indeed a form of embedded EM training [16], and more generally is a specific form of EM [18]. Our approach is similar to [14], except our underlying language models are backoff-based and thus retain the benefits of advanced smoothing methods, and we utilize both a normal and a backoff EM step as will be seen. We moreover wrap up the above ideas in the framework of dynamic Bayesian networks, which are used to represent and train all of our models.

We evaluate our methods on the ICSI meeting recorder dialog act (MRDA) [24] corpus, and find that our novel hidden backoff model can significantly improve dialog tagging accuracy. With a different number of hidden states for each dialog act, a relative reduction in tagging error rate as much as 6.1% can be achieved. Our best result shows an accuracy that is competitive with the best known result in this corpora but that also uses acoustic prosody as a feature (which we do not yet employ). Furthermore, our results also show the effectiveness of our embedded EM procedure, as we demonstrate that it increases training log likelihoods, while simultaneously reducing error rate.

Section 2 briefly summarizes our baseline DBN-based models for DA tagging tasks. In Section 3, we introduce our HBMs. Section 4 contains experimental evaluations on the MRDA corpus and finally Section 5 concludes.

2 DBN-based Models for Tagging

Dynamic Bayesian networks (DBNs) [17] are widely used in sequential data analysis such as automatic speech recognition (ASR) and DNA sequencing analysis [6]. A hidden Markov model (HMM) for dialog act tagging as in [27] is one such instance.

Figure 1 shows a generative DBN model that will be taken as our baseline. This DBN shows a prologue (the first time slice of the model), an epilogue (the last slice), and a chunk that is repeated sufficiently to fit the entire data stream. In this case, the data stream consists of the words of a meeting conversation, where individuals within the meeting (hopefully) take turns speaking. In our model, the entire meeting conversation, and all turns of all speakers, are strung together into a single stream rather than treating each turn in the meeting individually. This approach has the benefit that we are able to integrate a temporal DA-to-DA model (such as a DA bigram).

In all our models, to simplify we assume that the sentence change information is known (as is common with this corpus [24]). We next describe Figure 1 in detail. Normally, the *sentence change* variable is not

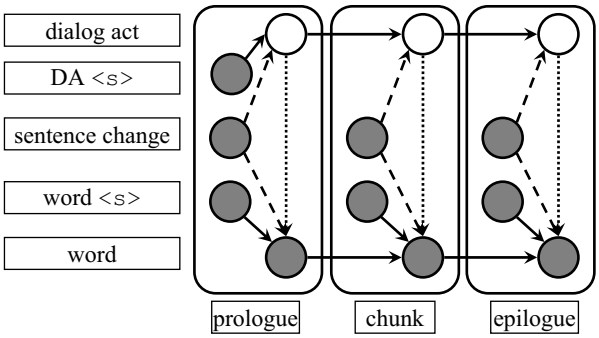


Figure 1: Baseline generative DBN for DA tagging.

set, so that we are within a sentence (or a particular DA). When a sentence change does not occur, the DA stays the same from slice to slice. During this time, we use a DA-specific language model (implemented via a backoff strategy) to score the words within the current DA.

When a sentence change event does occur, a new dialog act is predicted based on the dialog act from the previous sentence (using a DA bigram). At the beginning of a sentence, rather than conditioning on the last word of the previous sentence, we condition on the special start of sentence <s> token, as shown in the figure by having a special parent that is used only when *sentence change* is true. Lastly, at the very beginning of a meeting, a special start of DA token is used.

The joint probability under this baseline model is written as follows:

$$P(W, D) = \prod_k P(d_k|d_{k-1}) \cdot \prod_i P(w_{k,i}|w_{k,i-1}, d_k), \tag{1}$$

where $W = \{w_{k,i}\}$ is the word sequence, $D = \{d_k\}$ is the dialog act sequence, d_k is the dialog act of the k-th sentence, and $w_{k,i}$ is the i-th word of the k-th sentence in the meeting.

Because all variables are observed during training in our baseline case, we use the SRILM toolkit [25] with modified Kneser-Ney smoothing [5] to train all the models. In evaluations, the Viterbi algorithm [29] can be used to find the best dialog act sequence path from the words of the meeting according to the joint distribution in Equation (1).

3 Hidden Backoff Models

Structured language models [4] have shown impressive results on speech language tasks. In the case of analyzing discourse patterns, sentences with different dialog acts usually have different internal structures. In this work, while we do not use different structured language models for different dialog acts, we neither assume sentences for each dialog act will have the same hidden state patterns. For instance (and as mentioned above), a statement can consist of a noun followed by a verb phase.

Sub-DAs are not annotated in our training corpus. Moreover, clustering and annotation of these phrases is already a widely developed research topic [19, 13, 7]. In our approach, we use an EM algorithm to learn these hidden sub-DAs in a data-driven fashion. Pictorially, we add a layer of hidden states to our baseline DBN as illustrated in Figure 2.

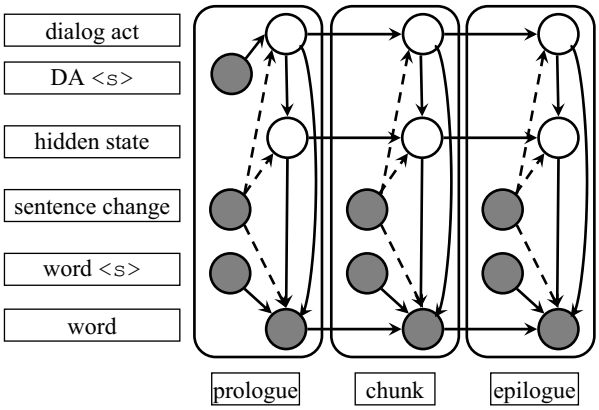


Figure 2: Hidden backoff model for DA tagging.

Under this model, the joint probability is:

$$\begin{aligned} P(W,S,D) = & \prod_k P(d_k|d_{k-1}) \\ & \cdot \prod_i \left[P(s_{k,i}|s_{k,i-1},d_k)\right. \\ & \left. \cdot\ P(w_{k,i}|w_{k,i-1},s_{k,i},d_k)\right], \end{aligned} \tag{2}$$

where $S = \{s_{k,i}\}$ is the hidden state sequence, $s_{k,i}$ is the hidden state at the i-th position of the k-th sentence, and other variables are the same as before.

Similar to our baseline model, the dialog act bigram $P(d_k|d_{k-1})$ can be modeled using a backoff bigram. Moreover, if the hidden states $\{s_{k,i}\}$ are known, the word prediction probability $P(w_{k,i}|w_{k,i-1}, s_{k,i}, d_k)$ can also use backoff and be trained accordingly. The hidden state sequence is unknown, however, and thus cannot be used to produce a backoff model. What we desire is an ability to utilize a backoff model (to mitigate data sparseness effects) while simultaneously retaining the state as a hidden (rather than an observed) variable, and also have a procedure that trains the entire model to improve overall model likelihood.

Expectation-maximization (EM) algorithms are well-known to be able to train models with hidden states. Furthermore, standard advanced smoothing methods such as modified Kneser-Ney smoothing [5] utilize integer counts (rather than fractional ones), and they moreover need "meta" counts (or counts of counts). Therefore, in order to train this model, we propose an embedded training algorithm that cycles between a standard EM training procedure (to train the hidden state distribution), and a stage where the most likely hidden states (and their counts and meta counts) are used externally to train a backoff model. This procedure can be described in detail as follows:

Input : W — meeting word sequence
Input : D — dialog act sequence
Output : $P(s_{k,i}|s_{k,i-1})$ - state transition CPT
Output : $P(w_{k,i}|w_{k,i-1}, s_{k,i}, d_k)$ - word model

randomly generates S;
train $P(w_{k,i}|w_{k,i-1}, s_{k,i}, d_k)$ backoff;
while *not "converged"* **do**
    train $P(s_{k,i}|s_{k,i-1})$ by EM;
    calculate best $\bar{S}$ sequence by Viterbi;
    train $P(w_{k,i}|w_{k,i-1}, \bar{s}_{k,i}, d_k)$ backoff;
end

Algorithm 1: Embedded training for hidden backoff models

In the algorithm, the input contains words and dialog act sequences of each sentence in the meeting. The output is the corresponding conditional probability table (CPT) for hidden state transitions, and a backoff model for word prediction. Because we train the backoff model when some of the variables are hidden, we call the result a *hidden backoff model*.

When decoding with such a hidden backoff model, a conventional Viterbi algorithm can be used to calculate the best dialog act sequence as follows:

$$D^* = \underset{D}{\text{argmax}}\ P(W, D) \approx \underset{D}{\text{argmax}}\ \max_{S} P(W, S, D),$$

where the sum is replaced by max (Viterbi assumption). While embedded Viterbi estimation is not guaranteed to have the same convergence (or fixed-point under convergence) as normal EM, we find empirically this to be the case (see below).

4 Experimental Results

We evaluated our hidden backoff model on the ICSI meeting recorder dialog act (MRDA) corpus [24]. MRDA is a rich data set that contains 75 natural meetings on different topics with each meeting involving about 6 participants. Dialog act annotations from ICSI were based on a previous approach in [10] with some adaptation for meetings in a number of ways described in [3]. Each dialog act contains a main tag, several optional special tags and an optional "disruption" form. The total number of distinct dialog acts in the corpus is as large as 1260. In order to make the problem comparable to other work [1], a dialog act tag sub-set is used in our experiments that contains back channels (b), place holders (h), questions (q), statements (s), and disruptions (x). In our evaluations, among the entire 75 conversations, 51 are used as the training set, 11 are used as the development set, 11 are used as test set, and the remaining 3 are not used.

Our baseline system is the generative model shown in Figure 1 and uses an backoff implementation of the word model, and is optimized on the development set. Our baseline system has an error rate of 19.7% on the test set, which is comparable to other approaches on the same task [1].

4.1 Same number of states for all dialog acts

To compare against our baseline, we use HBMs in the model shown in Figure 2. To train, we followed Algorithm 1 as described before and as is here detailed in Figure 3.

In this implementation, an upper triangular matrix (with self-transitions along the diagonal) is used for the hidden state transition probability table so that sub-DA states only propagate in one direction. When initializing the hidden state sequence of a DA, we expanded the states uniformly along the sentence. This

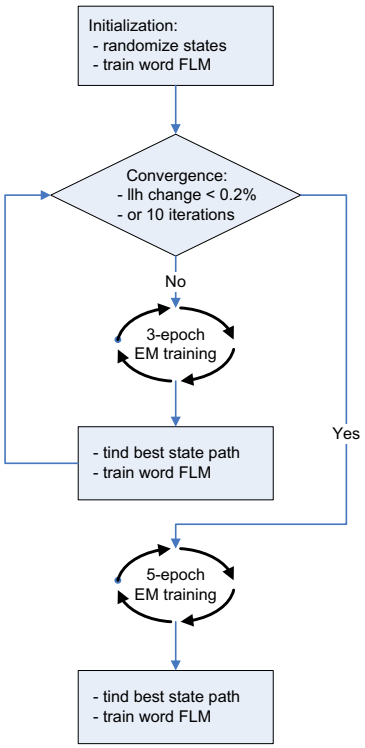


Figure 3: Embedded training: llh = log likelihood

initial alignment is then used for HBM training. In the word models used in our experiments, the backoff path first drops previous words, then does a parallel backoff to hidden state and dialog act using a mean combination strategy.

The HBM thus obtained was then fed into the main loop of our embedded EM algorithm. The training was considered to have "converged" if either it exceeded 10 iterations (which never happened) or the relative log likelihood increase was less than 0.2%. Within each embedded iteration, three EM epochs were used. After each EM iteration, a Viterbi alignment was performed thus obtaining what we expect to be a better hidden state alignment. This updated alignment, was then used to train a new HBM. The newly generated model was then fed back into the embedded training loop until it converged. After the procedure met our convergence criteria, an additional five EM epochs were carried out in order to provide a good hidden state transition probability table. Finally, after Viterbi alignment and text generation was performed, the word HBM was trained from the best state sequence.

To evaluate our hidden backoff model, the Viterbi algorithm was used to find the best dialog act sequence according to test data, and the tagging error rates were calculated. In our first experiment, an equal number of hidden states for all dialog acts were used in each model. The effect of this number on the accuracy of dialog act tagging are shown in Table 1.

Table 1: Hidden backoff models with different numbers of hidden states.

# states	error	improvement
baseline	19.7%	–
2-state	18.7%	5.1%
3-state	19.5%	1.0%

From Table 1 we see that with two hidden states for every dialog act the system can reduce the tagging error rate by more than 5% relative. As a comparison, in [1], where conditional maximum entropy models are used, the error rate is 18.8% when using both word and acoustic prosody features, and and 20.5% without prosody. Our results do not (yet) use prosody information. When the number of hidden states increases to 3, the improvement decreases even though it is still (very slightly) better than the baseline. We believe the reasons are as follows: First, assuming different dialog acts have the same number of hidden states may not be appropriate. For example, back channels usually have shorter sentences and are constant in discourse pattern over a DA. On the other hand, questions and statements typically have longer, and more complex, discourse structures. Second, even under the same dialog act, the structure and inherent length of sentence can vary. For example, "yes" can also be a statement even though it has only one word. Therefore, one-word statements need completely different hidden state patterns than those in subject-verb-object like statements — having one monolithic 3-state model for statements might be inappropriate. This issue is discussed further in Section 4.4.

4.2 Different number of states for different dialog acts

In order to mitigate the first problem described above, we allow different numbers of hidden states for each dialog act. This, however, leads to a combinatorial explosion of possibilities if done in a naïve fashion. Therefore, we attempted only a small number of combinations based on the statistics of numbers of words in each dialog act given in Table 2.

Table 2: Length statistics of different dialog acts.

DA	mean	median	std	p
(b)	1.0423	1	0.2361	0.4954
(h)	1.3145	1	0.7759	0.4660
(q)	6.5032	5	6.3323	0.3377
(s)	8.6011	7	7.8380	0.3013
(x)	1.7201	1	1.1308	0.4257

Table 2 shows the mean and median number of words per sentence for each dialog act as well as the standard deviation. Also, the last column provides the p value according to fitting the length histogram to a geometric distribution $(1-p)^n p$. As we expected, back channels (b) and place holders (h) tend to have shorter sentences while questions (q) and statements (s) have longer ones. From this analysis, we use fewer states for (b) and (h) and more states for (q) and (s). For disruptions (x), the standard deviation of number of words histogram is relatively high compared with (b) and (h), so we also used more hidden states in this case. In our experimental results below, we used one state for (b) and (h), and various numbers of hidden states for other dialog acts. The tagging error rates are shown in Table 3.

Table 3: Different numbers of hidden states for different dialog acts.

b	h	q	s	x	error	improvement
1	1	4	4	1	18.9%	4.1%
1	1	3	3	2	18.9%	4.1%
1	1	2	2	2	18.7%	5.1%
1	1	3	2	2	18.6%	5.6%
1	1	3	2	2	18.5%	6.1%

From Table 3, we see that using different numbers of hidden states for different dialog acts can produce better models. Among all the experiments we performed, the best case is given by three states for (q), two states for (s) and (x), and one state for (b) and (h). This combination gives 6.1% relative reduction of error rate from the baseline.

4.3 Effect of embedded EM training

Incorporating backoff smoothing procedures into Bayesian networks (and hidden variable training in particular) can show benefits for any data domain where smoothing is necessary. To understand the properties of our algorithm a bit better, after each training iteration using a partially trained model, we calculated both the log likelihood of the training set and the tagging error rate of the test data. Figure 4 shows these results using the best configuration from the previous section (three states for (q), two for (s)/(x) and one for (b)/(h)). This example is typical of the convergence we see of Algorithm 1, which empirically suggests that our procedure may be similar to a generalized EM [18].

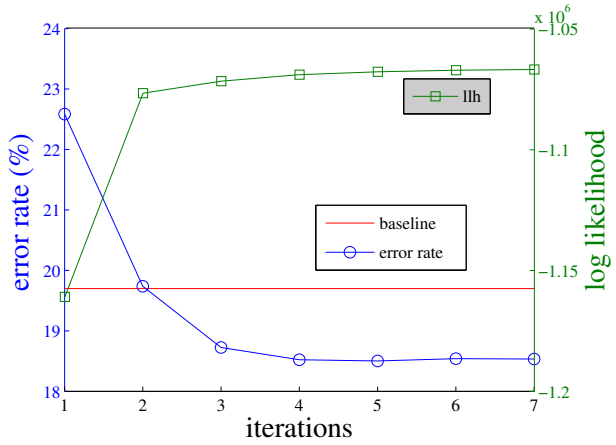


Figure 4: Embedded EM training performance.

First, we find that the log likelihood after each EM training is strictly increasing. This means that our embedded EM algorithm for hidden backoff models is effectively improving the overall joint likelihood of the training data according to the model. This strict increase of likelihood combined with the fact that Viterbi training does not have the same theoretical convergence guarantees as does normal EM indicates that more detailed theoretical analysis of this algorithm used with these particular models is desirable.

From the figure we also see that both the log likelihood and tagging error rate "converge" after around four iterations of embedded training. This quick convergence indicates that our embedded training procedure is effective. The leveling of the error rates after several iterations shows that model over-fitting appears not to be an issue in our case presumably due to the smoothed embedded backoff models.

4.4 Discussion and Error Analysis

A large portion of our tagging errors are due to confusing the DA of short sentences such as "yeah", and "right". The sentence, "yeah" can either be a back channel or an affirmative statement. There are also cases where "yeah?" is a question. These types of confusions are difficult to solve in our framework (without using acoustic prosody) but there are several possibilities. First, we can allow the use of a "fork and join" transition matrix, where we fork to each DA-specific condition (e.g., short or long) and join thereafter. Alternatively,

hidden Markov chain structuring algorithms can be used [26]. Also, context (i.e., conditioning the number of sub-DAs on the previous DA) might be helpful.

More promisingly, research has shown that acoustic prosody can yield large improvements in dialog act tagging tasks [23]. We believe that our results too can be improved by incorporating prosody information since we expect a different pitch contour for statements, questions, and back channels. We could also get information from speaker turns which we already have labeled in the corpus. For an example, if the speaker before "yeah" and after are the same one, then there is a high chance that this "yeah" is a back channel. We plan to try both of the above in future work. We also plan to evaluate models where sentence change is a hidden variable used in an HBM.

Finding a proper number of hidden states for each dialog act is also challenging. In our preliminary work, we simply explored different combinations using simple statistics of the data. A systematic procedure would be more beneficial. In this work, we also did not perform any hidden state tying within different dialog acts. In practice, some states in statements should be able to be beneficially tied with other states within questions. Our results show that having three states for all dialog acts is not as good as two states for all. But with tying, more states might be more successfully used.

5 Conclusions

In this work, we proposed a procedure we call hidden backoff models to solve the problem in dialog act tagging, where we wish to use smoothing backoff models that involve variables that are intrinsically hidden at training time. Moreover, we tested this procedure in the context of dynamic Bayesian networks (DBNs). Different hidden states were used to model different positions in a dialog act. Because smoothing techniques are required in language model training when vocabulary sizes are large, we proposed an embedded EM training algorithm to train such hidden backoff models.

According to empirical evaluations, our embedded EM algorithm effectively increases log likelihood on training data and reduces DA tagging error rate on test data. We have also shown that our hidden backoff model can reduce the dialog act tagging error rate. If different numbers of hidden states are used in different dialog acts, we can find a better combination that reduces the tagging error rate by 6.1% relative to the baseline.

References

[1] J. Ang, Y. Liu, and E. Shriberg. Automatic dialog act segmentation and classification in multiparty meetings. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Philadelphia, PA, March 2005.

[2] P. Bartlett, M. Collins, B. Taskar, and D. McAllester. Exponentiated gradient algorithms for large-margin structured classification. In *Advances in Neural Information Processing Systems (NIPS)*, Vancouver, Canada, 2004.

[3] S. Bhagat, R. Dhillon, H. Carvey, and E. Shriberg. Labeling guide for dialog act tags in the meeting recordering meetings. Technical Report 2, International Computer Science Insititute, Berkeley, August 2003.

[4] C. Chelba and F. Jelinek. Recognition performance of a structured language model. In *Proceedings of Eurospeech*, September 1999.

[5] Stanley F. Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Computer Science Group, Harvard University, August 1998.

[6] Richard Durbin, Sean R. Eddy, Anders Krogh, and Graeme Mitchison. *Biological Sequence Analysis : Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1999.

[7] D. Gildea and D. Jurafsky. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288, September 2002.

[8] Y. He and S. Young. A data-driven spoken language understanding system. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 583–588, 2003.

[9] D. Jurafsky, R. Bates, N. Coccaro, R. Martin, Marie Meteer, Klaus Ries, Elizabeth Shriberg, Andreas Stolcke, Paul Taylor, and Carol Van Ess-Dykema. Automatic detection of discourse structure for speech recognition and understanding. In *Proceedings of the 1997 IEEE Workshop on Speech Recognition and Understanding*, 1997.

[10] D. Jurafsky, E. Shriberg, and D. Biasca. Switchboard SWBD-DAMSL shallow-discourse-function annotation coders manual. Technical Report 97-02, Institute of Cognitive Science, University of Colorado, 1997.

[11] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning*, 2001.

[12] H. Lee, J.-W. Lee, and J. Seo. Speech act analysis model of Korean utterances for automatic dialog translation. *Journal of KISS(B) (Software and Applications)*, 25(10):1443–1452, 1998.

[13] Kong Joo Lee, Gil Chang Kim, Jae-Hoon Kim, and Y. S. Han. Restricted representation of phrase structure grammar for building a tree annotated corpus of Korean. *Natural Language Engineering*, 3(2-3):215–230, September 1997.

[14] Kristine W. Ma, George Zavaliagkos, and Marie Meteer. Bi-modal sentence structure for language modeling. *Speech Communication*, 31(1):51–67, May 2000.

[15] Marion Mast, Heinrich Niemann, Elmar Nöth, and Ernst Günter Schukat-Talamazzini. Automatic classification of dialog acts with semantic classification trees and polygrams. *Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing*, pages 217–229, 1996.

[16] N. Morgan and H. Bourlard. Continuous speech recognition using multilayer perceptrons with hidden Markov models. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 413–416, 1990.

[17] Kevin Murphy. *Dynamic Bayesian Networks, Representation, Inference, and Learning*. PhD thesis, MIT, Department of Computer Science, 2002.

[18] Radford M. Neal. A view of the EM algorithm that justifies incremental, sparse, and other variants. In Michael I. Jordan, editor, *Learning in Graphical Models*, pages 355–368. Dordrecht: Kluwer Academic Publishers, 1998.

[19] R. Pieraccini and E. Levin. Stochastic representation of semantic structure for speech understanding. In *2nd European Conference on Speech Communication and Technology Proceedings*, volume 2, pages 383–386, 1991.

[20] N. Reithinger, R. Engel, M. Kipp, and M. Klesen. Predicting dialogue acts for a speech-to-speech translation system. In *Proceedings of the International Conference on Spoken Language Processing*, pages 654–657, October 1996.

[21] N. Reithinger and M. Klesen. Dialogue act classification using language models. In *Proceedings of Eurospeech*, September 1997.

[22] J. R. Searle. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, 1969.

[23] E. Shriberg, R. Bates, A. Stolcke, P. Taylor, D. Jurafsky, K. Ries, N. Coccaro, R. Martin, M. Meteer, and C. van Ess-Dykema. Can prosody aid the automatic classification of dialog acts in conversational speech? *Language and Speech*, 41(3–4):439–487, 1998.

[24] Elizabeth Shriberg, Raj Dhillon, Sonali Bhagat, Jeremy Ang, and Hannah Carvey. The ICSI meeting recorder dialog act (MRDA) corpus. In *Proc. of the 5th SIGdial Workshop on Discourse and Dialogue*, pages 97–100, 2004.

[25] Andreas Stolcke. SRILM – an extensible language modeling toolkit. In *Proceedings of International Conference on Spoken Language Processing*, volume 2, pages 901–904, Denver, CO, September 2002.

[26] Andreas Stolcke and Stephen M. Omohundro. Hidden Markov model induction by bayesian model merging. In Stephen Jose Hanson, Jack D. Cowan, and C. Lee Giles, editors, *Advances in Neural Information Processing Systems*, pages 11–18, Denver, Colorado, November 1992. Morgan Kaufmann.

[27] Andreas Stolcke, Elizabeth Shriberg, Rebecca Bates, Noah Coccaro, Daniel Jurafsky, Rachel Martin, Marie Meteer, Klaus Ries, Paul Taylor, and Carol Van Ess-Dykema. Dialog act modeling for conversational speech. In *Proc. of the AAAI Spring Symp. on Appl. Machine Learning to Discourse Processing*, pages 98–105, 1998.

[28] C. Sutton, K. Rohanimanesh, and A. McCallum. Dynamic conditional random fields: factorized probabilistic models for labeling and segmenting sequence data. In *Proc. of ICML*, 2004.

[29] A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, 1967.