Discriminative Hidden Markov Modeling with Long State Dependence using a kNN Ensemble

ZHOU GuoDong
Institute for Infocomm Research
21 Heng Mui Keng Terrace
Singapore 119613
Email: zhougd@i2r.a-star.edu.sg

Abstract

This paper proposes a discriminative HMM (DHMM) with long state dependence (LSD-DHMM) to segment and label sequential data. The LSD-DHMM overcomes the strong context independent assumption in traditional generative HMMs (GHMMs) and models the sequential data in a discriminative way, by assuming a novel mutual information independence. As a result, the LSD-DHMM separately models the long state dependence in its state transition model and the observation dependence in its output model. In this paper, a variable-length mutual information-based modeling approach and an ensemble of kNN probability estimators are proposed to capture the long state dependence and the observation dependence respectively. The evaluation on shallow parsing shows that the LSD-DHMM not only significantly outperforms GHMMs but also much outperforms other DHMMs. This suggests that the LSD-DHMM can effectively capture the long context dependence to segment and label sequential data.

1. Introduction

A Hidden Markov Model (HMM) is a model where a sequence of observations is generated in addition to the Markov state sequence. It is a latent variable model in the sense that only the observation sequence is known while the state sequence remains “hidden”. In recent years, HMMs have enjoyed great success in many tagging applications, most notably part-of-speech (POS) tagging (Church 1988; Weischedel et al 1993; Merialdo 1994) and named entity recognition (Bikel et al 1999; Zhou et al 2002). Moreover, there have been also efforts to extend the use of HMMs to word sense disambiguation (Segond et al 1997) and shallow/full parsing (Brants et al 1997; Skut et al 1998; Zhou et al 2000).

Traditionally, a HMM segments and labels sequential data in a generative way, assigning a joint probability to paired observation and state sequences. More formally, a generative (first-order) HMM (GHMM) is given by a finite set of states S including an designated initial state and an designated final state, a set of possible observation O, two conditional probability distributions: a state transition model from $s^{'}$ to s, $p(s \mid s^{'})$ for $s^{'}, s \in S$ and an output model, $p(o \mid s)$ for $o \in O, s \in S$. A sequence of observations is generated by starting from the designated initial state, transmiting to a new state according to $p(s \mid s^{'})$, emitting an observation selected by that new state according to $p(o \mid s)$, transmiting to another new state and so on until the designated final state is generated.

There are several problems with this generative approach. First, many tasks would benefit from a richer representation of observations—in particular a representation that describes observations in terms of many overlapping features, such as capitalization, word endings, part-of-speech in addition to the traditional word identity. Note that these features always depends on each other. Furthermore, to define a joint probability over the observation and state sequences, the generative approach needs to enumerate all the possible observation sequences. However, in some tasks, the set of all the possible observation sequences is not reasonably enumerable. Second, the generative approach fails to effectively model the dependence in the observation sequence. Moreover, it is difficult for the generative approach to model the long state dependence since it is not reasonably practical for ngram modeling(e.g. bigram for the first-order GHMM and trigram for the secnod-order GHMM) to be beyond trigram. Third, the generative approach normally estimates the parameters to maximize the likelihood of the observation sequence. However, in many NLP tasks, the goal is to predict the state sequence given the observation sequence. In other words, the generative approach inappropriately applies a generative joint probability model for a conditional probability problem. In summary, the main reasons behind these problems of the generative approach are the strong context independent assumption and the generative nature in modeling sequential data.

While the dependence between successive states can be directly modeled by its state transition model, the generative approach fails to directly capture the observation dependence in the output model. From this viewpoint, a GHMM can be also called an observation independent HMM.

To resolve above problems in GHMMs, some researches have been done to move from the generative approach to the discriminative approach. Discriminative HMMs (DHMMs) do not expend modeling effort on the observation sequnce, which are fixed at test time. Instead, DHMMs model the state sequence depending on arbitrary, non-independent features of the observation sequence, normally without forcing the model to account for the distribution of those dependencies. Punyakanok and Roth (2000) proposed a projection-based DHMM (PDHMM) which represents the probability of a state transition given not only the current observation but also past and future observations and used the SNoW classifier (Roth 1998, Carlson et al 1999) to estimate it (SNoW-PDHMM thereafter). McCallum et al (2000) proposed the extact same model and used maximum entropy to estimate it (ME-PDHMM thereafter). Lafferty et al (2001) extanded ME-PDHMM using conditional random fields by incorporating the factored state representation of the same model (that is, representing the probability of a state given the observation sequence and the previous state) to alleviate the label bias problem in projection-based DHMMs, which can be biased towards states with few successor states (CRF-DHMM thereafter). Similar work can also be found in Bouttou (1991). Punyakanok and Roth (2000) also proposed a non-projection-based DMM which separates the dependence of a state on the previous state and the observation sequence, by rewriting the GHMM in a discriminative way and heuristically extending the notation of an observation to the observation sequence. Zhou et al (2000) systematically derived the exact same model as in Punyakanok and Roth (2000) and used back-off modeling to esimate the probability of a state given the observation sequence (Backoff-DHMM thereafter) while Punyakanok and Roth (2000) used the SNoW classifier to estimate it(SNoW-DHMM thereafter).

This paper follows our previous work in Zhou et al (2000) and proposes an alternative non-projection-based DHMM with long state dependence (LSD-DHMM), which separates the dependence of a state on the previous states and the observation sequence. Moreover, a variable-length mutual information based modeling approach (VLMI) is proposed to capture the long state dependence of a state on the previous states. In addition, an ensemble of kNN probability estimators is proposed to capture the observation dependence of a state on the observation sequence. Experimentation shows that VLMI effectively captures the long state dependence. It also shows that the kNN ensemble captures the dependence between the features of the observation sequence more effectively than classifier-based approaches, by forcing the model to account for the distribution of those dependencies.

The layout of this paper is as follows. Section 2 first proposes the LSD-DHMM and then presents the VLMI to capture the long state dependence. Section 3 presents the kNN probability estimator to capture the observation dependence while Section 4 presents the kNN ensemble. Section 5 introduces shallow parsing, while experimental results are given in Section 6. Finally, some conclusion will be drawn in Section 7.

2. LSD-DHMM: Discriminative HMM with Long State Dependence

In principle, given an observation sequence $o_1^n = o_1 o_2 \cdots o_n$, the goal of a conditional probability model is to find a stochastic optimal state sequence $s_1^n = s_1 s_2 \cdots s_n$ that maximizes $\log p(s_1^n \mid o_1^n)$

$$S^* = \arg\max_{s_1^n} \log p(s_1^n \mid o_1^n) \quad (1)$$

By applying the Bayes' rule, we can rewrite the equation (1) as:

$$\begin{aligned} s^* &= \arg\max_{s_1^n} \{\log p(s_1^n \mid o_1^n)\} \\ &= \arg\max_{s_1^n} \{\log p(s_1^n) + MI(s_1^n, o_1^n)\} \end{aligned} \quad (2)$$

Obviously, the second term $MI(s_1^n, o_1^n)$ captures the mutual information between the state sequence s_1^n and the observation sequence o_1^n. To compute $MI(s_1^n, o_1^n)$ efficiently, we propose a novel mutual information independence assumption:

$$MI(s_1^n, o_1^n) = \sum_{i=1}^{n} MI(s_i, o_1^n) \quad \text{or}$$

$$\log \frac{p(s_1^n, o_1^n)}{p(s_1^n) \cdot p(o_1^n)} = \sum_{i=1}^{n} \log \frac{p(s_i, o_1^n)}{p(s_i) \cdot p(o_1^n)} \quad (3)$$

That is, we assume a state is only dependent on the observation sequence o_1^n and independent on other states in the state sequence s_1^n. This assumption is reasonable because the dependence among the states in the state sequence s_1^n has been directly captured by the first term $\log p(s_1^n)$ in equation (2).

By applying the assumption (3) into the equation (2) and using the chain rule, we have:

$$\begin{aligned}
s^* &= \arg\max_{s_1^n}\{\sum_{i=2}^{n}\log p(s_i \mid s_1^{i-1}) + \log p(s_1) \\
&\qquad - \sum_{i=1}^{n}\log p(s_i) + \sum_{i=1}^{n}\log p(s_i \mid o_1^n)\} \\
&= \arg\max_{s_1^n}\{\sum_{i=2}^{n}\log p(s_i \mid s_1^{i-1}) - \sum_{i=2}^{n}\log p(s_i) \\
&\qquad + \sum_{i=1}^{n}\log p(s_i \mid o_1^n)\} \\
&= \arg\max_{s_1^n}\{\sum_{i=2}^{n} MI(s_i, s_1^{i-1}) + \sum_{i=1}^{n}\log p(s_i \mid o_1^n)\}
\end{aligned} \tag{4}$$

The above model consists of two models: the state transition model $\sum_{i=2}^{n} MI(s_i, s_1^{i-1})$ which measures the state dependence of a state given the previous states, and the output model $\sum_{i=1}^{n}\log p(s_i \mid o_1^n)$ which measures the observation dependence of a state given the observation sequence in a discriminative way. Therefore, we call the above model as in equation (4) a discriminative HMM (DHMM) with long state dependence (LSD-DHMM). The LSD-DHMM separates the dependence of a state on the previous states and the observation sequence. The main difference between a GHMM and a LSD-DHMM lies in their output models in that the output model of a LSD-DHMM directly captures the context dependence between successive observations in determining the "hidden" states while the output model of the GHMM fails to do so. That is, the output model of a LSD-DHMM overcomes the strong context independent assumption in the GHMM and becomes observation context dependent. Therefore, the LSD-DHMM can also be called an observation context dependent HMM. Compared with other DHMMs, the LSD-DHMM explicitly models the long state dependence and the non-projection nature of the LSD-DHMM alleviates the label bias problem inherent in projection-based DHMMs.

Computation of a LSD-DHMM consists of two parts. The first is to compute the state transition model: $\sum_{i=2}^{n} MI(s_i, s_1^{i-1})$. Traditionally, ngram modeling(e.g. bigram for the first-order GHMM and trigram for the second-order GHMM) is used to estimate the state transition model. However, such approach fails to capture the long state dependence since it is not reasonably practical for ngram modeling to be beyond trigram. In this paper, a variable-length mutual information-based modeling approach (VLMI) is proposed as follow: For each $i(2 \le i \le n)$, we first find a minimal $k(0 \le k \prec i)$ where the frequency of s_k^{i-1} is bigger than a threshold (e.g. 10) and then estimate $MI(s_i, s_1^{i-1})$ using $MI(s_i, s_k^{i-1}) = \frac{p(s_k^i)}{p(s_i)\cdot p(s_k^{i-1})}$. In this way, the long state dependence can be captured maximally in a dynamical way. Here, the frequencies of variable-length state sequences are estimated using the simple Good-Turing approach (Gale et al 1995).

The second is to estimate the output model: $\sum_{i=1}^{n}\log p(s_i \mid o_1^n)$. Ideally, we would have sufficient training data for every event whose conditional probability we wish to calculate. Unfortunately, there is rarely enough training data to compute accurate probabilities when decoding on new data. Traditionally, there are two existing approaches to resolve this problem: linear interpolation (Jelinek 1989) and back-off (Katz 1987). However, these two approaches only work well when the number of different information sources is limited. When a long context is considered, the number of different information sources is exponential and not reasonably enumerable. The current tendency is to recast it as a classification problem and use the output of a classifier, e.g. the maximum entropy classifier (Ratnaparkhi 1999) to estimate the state probability distribution given the observation sequence. In the next two sections, we will propose a more effective ensemble of kNN probability estimators to resolve this problem.

3. kNN Probability Estimator

The main challenge for the LSD-DHMM is how to reliably estimate $p(s_i \mid o_1^n)$ in its output model. For efficiency, we can always assume $p(s_i \mid o_1^n) \approx p(s_i \mid E_i)$, where the pattern entry $E_i = o_{i-N} \cdots o_i \cdots o_{i+N}$. That is, we only consider the observation dependence in a window of 2N+1 observations (e.g. we only consider the current observation, the previous observation and the next observation when N=1). For convenience, we denote $P(\bullet \mid E_i)$ as the conditional state probability distribution of the states given E_i and

$p(s_i \mid E_i)$ as the conditional state probability of s_i given E_i.

The kNN probability estimator estimates $P(\bullet \mid E_i)$ by first finding the K nearest neighbors of frequently occurring pattern entries $kNN(E_i) = \{E_i^k \mid k = 1,2,...,K\}$ and then aggregating them to make a proper estimation of $P(\bullet \mid E_i)$. Here, the conditional state probability distribution is estimated instead of the classification in a traditional kNN classifier. To do so, all the frequently occurring pattern entries are extracted from the training corpus in an exhaustive way and stored in a dictionary *FrequentEntryDictionary*. In order to limit the dictionary size and keep efficiency, we constrain a valid set of pattern entry forms *ValidEntryForm* to consider only the most informative information sources. Generally, *ValidEntryForm* can be determined manually or automatically according to the applications. In Section 5, we will give an example.

Given a pattern entry E_i and a dictionary of frequently occurring pattern entries *FrequentEntryDictionary*, a simple algorithm is applied to find the K nearest neighbors of the pattern entry E_i from the dictionary as follows:

- compare E_i with each entry in the dictionary and find all the compatible entries
- compute the cosine similarity between E_i and each of the compatible entries
- sort out the K nearest neighbors according to their cosine similarities

Finally, the conditional state probability distribution of the pattern entry is aggregated over those of its K nearest neighbors weighted by their frequencies $f(E_i^k)$ and cosine similarities $\hat{p}(E_i^k \mid kNN)$:

$$P(\bullet \mid E_i) = \frac{\sum_{k=1}^{K} \hat{p}(E_i^k \mid kNN) \cdot f(E_i^k) \cdot P(\bullet \mid E_i^k)}{\sum_{k=1}^{K} \hat{p}(E_i^k \mid kNN) \cdot f(E_i^k)} \tag{5}$$

4. kNN Ensemble

In the literature, an ensemble has been widely used in the classification problem to combine several classifiers (Breiman 1996; Hamamoto 1997; Dietterich 1998; Zhou Z.H. et al 2002; Kim et al 2003). It is well known that an ensemble often outperforms the individual classifiers that make it up (Hansen et al 1990).

In this paper, an ensemble of kNN probability estimators is proposed to estimate the conditional state probability distribution $P(\bullet \mid E_i)$ instead of the classification. This is done through a bagging technique (Breiman 1996) to aggregate several kNN probability estimators. In bagging, the M kNN probability estimators in the ensemble $ENS = \{kNN_m \mid m = 1,2,...,M\}$ are trained independently via a bootstrap technique and then they are aggregated via an appropriate aggregation method. Usually, we have a single training set and need M training sample sets to construct a kNN ensemble with M independent kNN probability estimators. From the statistical viewpoint, we need to make the training sample sets different as much as possible in order to obtain a higher aggregation performance. For doing this, we often use the bootstrap technique which builds M replicate data sets by randomly re-sampling with replacement from the given training set repeatedly. Each example in the given training set may appear repeatedly or not at all in any particular replicate training sample set. Each training sample set is used to train a certain kNN probability estimator. Finally, the conditional state probability distribution of the pattern entry E_i is averaged over those of the M kNN probability estimators in the ensemble:

$$P(\bullet \mid E_i) = \frac{\sum_{m=1}^{M} P(\bullet \mid E_i, kNN_m)}{M} \tag{6}$$

5. Shallow Parsing

In order to evaluate the LSD-DHMM and the proposed variable-length mutual information modeling approach for the long state dependence in the state transition model and the kNN ensemble for the observation dependence in the output model, we have applied it in the application of shallow parsing.

For shallow parsing, we have $o_1 = p_i w_i$, where $w_1^n = w_1 w_2 \cdots w_n$ is the word sequence and $p_1^n = p_1 p_2 \cdots p_n$ is the part-of-speech (POS) sequence, while the "hidden" states are represented as structural tags to bracket and differentiate various categories of phrases. The basic idea of using the structural tags to represent the "hidden" states is similar to Skut et al (1998) and Zhou et al (2000). Here, a structural tag consists of three parts:

- **Boundary Category (BOUNDARY)**: it is a set of four values: "O"/"B"/"M"/"E", where "O" means that current word is a wh**O**le phrase and "B"/"M"/"E" means that current word is at the **B**eginning/in the **M**iddle/at the **E**nd of a phrase.
- **Phrase Category (PHRASE)**: it is used to denote the category of the phrase.
- **Part-of-Speech (POS)**: Because of the limited number of boundary and phrase categories, the POS is added into the structural tag to represent more accurate state transition and output models.

For example, given the following POS tagged sentence as the observation sequence:

He/PRP reckons/VBZ the/DT current/JJ account/NN deficit/NN will/MD narrow/VB to/TO only/RB $/$ 1.8/CD billion/CD in/IN September/NNP ./.

We can have a corresponding sequence of structural tags as the "hidden" state sequence:

O_NP_PRP(He/PRP) O_VP _VBZ (reckons/VBZ) B_NP _DT (the/DT) M_NP _JJ (current/JJ) M_NP _NN (account/NN) E_NP _NN (deficit/NN) B_VP _MD (will/MD) E_VP _VB (narrow/VB) O_PP _TO (to/TO) B_QP _RB (only/RB) M_QP _$ ($/$) M_QP _CD (1.8/CD) E_QP _CD (billion/CD) O_PP _IN (in/IN) O_NP _NNP(September/NNP) O_O _. (./.)

and an equivalent phrase chunked sentence as the shallow parsing result:

[NP He/PRP] [VP reckons/VBZ] [NP the/DT current/JJ account/NN deficit/NN] [VP will/MD narrow/VB] [PP to/TO] [QP only/RB $/$ 1.8/CD billion/CD] [PP in/IN] [NP September/NNP] [O ./.]

6. Experimentation

The corpus used in shallow parsing is extracted from the PENN TreeBank (Marcus et al. 1993) of 1 million words (25 sections) by a program provided by Sabine Buchholz from Tilburg University. All the evaluations are 5-fold cross-validated. For shallow parsing, we use the F-measure to measure the performance. Here, the F-measure is the weighted harmonic mean of the precision (P) and the recall (R): $F = \frac{(\beta^2 + 1)RP}{\beta^2 R + P}$ with β^2=1 (Rijsbergen 1979), where the precision (P) is the percentage of predicted phrase chunks that are actually correct and the recall (R) is the percentage of correct phrase chunks that are actually found.

Tables 1, 2 and 3 show the detailed performance of LSD-DHMMs. In this paper, the valid set of pattern entry forms $ValidEntryForm$ is defined to include those pattern entry forms within a windows of 7 observations(including current, left 3 and right 3 observations) where for w_j to be included in a pattern entry, all or one of the overlapping features in each of $p_j, p_{j+1} \cdots, p_i (j \le i)$ or $p_i, p_{i+1} \cdots, p_j (i \le j)$ should be included in the same pattern entry while for p_j to be included in a pattern entry, all or one of the overlapping features in each of $p_{j+1}, p_{j+2} \cdots, p_i (j \prec i)$ or $p_i, p_{i+1} \cdots, p_{j-1} (i \prec j)$ should be included in the same pattern entry.

Table 1 shows the effect of different number of nearest neighbors in the kNN probability estimator and considered previous states in the variable-length mutual information modeling approach of the LSD-DHMM, using only one kNN probability estimator in the ensemble to estimate $p(s_i \mid o_1^n)$ in the output model. It shows that finding 3 nearest neighbors in the kNN probability estimator performs best. It also shows that further increasing the number of nearest neighbors does not increase or even decrease the performance. This may be due to introduction of noisy neighbors when the number of nearest neighbors increases. Moreover, Table 1 shows that the LSD-DHMM performs best when six previous states is considered in the variable-length mutual information-based modeling approach and further considering more previous states only slightly increase the performance. This suggests that the state dependence exists well beyond traditional ngram modeling (e.g. bigram and trigram) to six previous states and the variable-length mutual information-based modeling approach can capture the long state dependence. In the following experimentation, we will only use the LSD-DHMM with 3 nearest neighbors used in the kNN probability estimator and 6 previous states considered in the variable-length mutual information modeling approach.

Table 2 shows the effect of different number of kNN probability estimators in the ensemble. It shows that 15 bootstrap replicates are enough for the k-NN ensemble on shallow parsing and increase the F-measure by 0.71 compared with the ensemble of only one kNN probability estimator.

Table 3 compares the LSD-DHMM with GHMMs and other DHMMs. It shows that all the DHMMs significantly outperform GHMMs due to the modeling of the observation dependence and allowing for non-independent, difficult to enumerate observation features. It also shows that our LSD-DHMM much outperforms other DHMMs due to the modeling of the long state dependence using the variable-length mutual information-based modeling approach in the LSD-DHMM. Moverover, Table 3 shows that no-projection-based DHMMs (i.e. CRF-DHMM, SNoW-DHMM, Backoff-DHMM and LSD-DHMM) outperform projection-based DHMMs. It may be due to alleviation of the label bias problem inherent in the projection-based DHMMs. Finally, Table 2 also compares the kNN ensemble with popular classifier-based approaches, such as SNoW and Maximum Entropy, in estimating the output model of the LSD-DHMM. It shows that the kNN ensemble outperforms these classifier-based approaches. This suggests that the kNN ensemble captures the dependence between the features of the observation sequence more effectively by forcing the model to account for the distribution of those dependencies.

Table 1: Effect of different numbers of nearest neighbors in the kNN probability estimator and previous states considered in the variable-length mutual information modeling approach of the LSD-DHMMs, using only a probability estimator in the ensemble

Shallow Parsing		Number of nearest neighbors				
		1	2	3	4	5
Number of considered previous states	1	93.12	93.50	93.76	93.70	93.66
	2	93.65	93.82	94.23	94.19	94.12
	4	93.90	94.15	94.42	94.38	94.35
	6	94.12	94.28	**94.53**	**94.54**	94.51
	8	94.15	94.35	**94.55**	94.52	94.50

Table 2: The Effect of different number of kNN probability estimators in the ensemble on shallow parsing

Number of kNN probability estimators in the ensemble	F-measure
1	94.53
2	94.77
4	94.93
8	95.06
14	95.21
15	**95.24**
16	95.24
20	95.25
25	95.25
28	95.36

Table 3: Comparison of LSD-DHMMs with GHMMs and other DHMMs

Models		F
GHMMs	First order	92.14
	Second order	92.41
DHMMs	ME-PDMM	93.26
	CRF-DMM	94.04
	SNoW-PDMM	93.44
	SNoW-DMM	94.12
	Backoff-DMM	93.68
	LSD-DMM(Ensemble)	**95.24**
	LSD-DMM(ME)	94.25
	LSD-DMM(SNoW)	94.41

7. Conclusion

Hidden Markov Models (HMMs) are a powerful probabilistic tool for modeling sequential data and have been applied with success to many text-related tasks, such as shallow paring. In these cases, the observations are usually modified as multinomial distributions over a discrete dictionary and the HMM parameters are set to maximize the likelihood of the observations. This paper presents a discriminative HMM with long state dependence that allows observations to be represented as arbitrary overlapping features and defines the conditional probability of the state sequence given the observation sequence. It does so by assuming a novel mutual information independence to separate the dependence of a state given the observation sequence and the previous states. Finally, the long state dependence and the observation dependence can be effectively captured by a variable-length mutual information model and a kNN ensemble respectively.

In future work, we will explore our model in other applications, such as full parsing.

References

Bikel D.M., Schwartz R. & Weischedel R.M. (1999). An Algorithm that Learns What's in a Name. *Machine Learning* (Special Issue on NLP). 34(3): 211-231.

Bottou L. (1991). Une approche theorique de l'apprentissage connexionniste: Applications a la reconnaissance de la parole. *Doctoral dissertation, Universite de Paris XI.*

Brants T., Skut W., & Krenn B. (1997). Tagging Grammatical Functions. *Proceedings of the Conference on Empirical Methods on Natural Language Processing (EMNLP'1997)*. Brown Univ. RI.

Carlson A, Cumby C. Rosen J. and Roth D. 1999. The SNoW learning architecture. Techinical

Report UIUCDCS-R-99-2101. UIUC Computer Science Department.

Church K.W. (1998). A Stochastic Pars Program and Noun Phrase Parser for Unrestricted Text. *Proceedings of the Second Conference on Applied Natural Language Processing (ANLP'1998).* Austin, Texas.

Fausett L. (1994). *Fundamentals of neural networks*. Prentice Hall Press.

Gale W.A. and Sampson G. 1995. Good-Turing frequency estimation without tears. *Journal of Quantitative Linguistics*. 2:217-237.

Jelinek F. (1989). Self-Organized Language Modeling for Speech Recognition. In Alex Waibel and Kai-Fu Lee(Editors). *Readings in Speech Recognitiopn*. Morgan Kaufmann. 450-506.

Katz S.M. (1987). Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. *IEEE Transactions on Acoustics. Speech and Signal Processing.* 35: 400-401.

Lafferty J. McCallum A and Pereira F. (2001). Conditional random fields: probabilistic models for segmenting and labeling sequence data. *ICML-20.*

Marcus M., Santorini B. & Marcinkiewicz M.A. (1993). Buliding a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*. 19(2):313-330.

McCallum A. Freitag D. and Pereira F. 2000. Maximum entropy Markov models for information extraction and segmentation. ICML-19. 591-598. Stanford, California.

Merialdo B. (1994). Tagging English Text with a Probabilistic Model. *Computational Linguistics.* 20(2): 155-171.

Punyakanok V. and Roth D. (2000). The Use of Classifiers in Sequential Inference *NIPS-13.*

Rabiner L.R. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition". *Proceedings of the IEEE,* 77(2): 257-286.

Ratnaparkhi A. 1999. Learning to parsing natural language with maximum entropy models. *Machine Learning*. 34:151-175.

Roth D. 1998. Learning to resolve natural language ambiguities: A unified approach. In *Proceedings of the National Conference on Artificial Intelligence.* 806-813.

Segond F., Schiller A., Grefenstette & Chanod F.P. (1997). An Experiment in Semantic Tagging using Hidden Markov Model Tagging. *Proceedings of the Joint ACL/EACL workshop on Automatic Information Extraction and Building of Lexical Semantic Resources.* pp.78-81. Madrid, Spain.

Skut W. & Brants T. (1998). Chunk Tagger – Statistical Recognition of Noun Phrases. *Proceedings of the ESSLLI'98 workshop on Automatic Acquisition of Syntax and Parsing.* Univ. of Saarbrucken. Germany.

van Rijsbergen C.J. (1979). *Information Retrieval.* Buttersworth, London.

Viterbi A.J. (1967). Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm. *IEEE Transactions on Information Theory.* 13: 260-269.

Weischedel R., Meteer M., Schwartz R., Ramshaw L. & Palmucci J. (1993). Coping with Ambiguity and Unknown Words through Probabilistic Methods. *Computational Linguistics.* 19(2): 359-382.

Zhou GuoDong & Su Jian, (2000). Error-driven HMM-based Chunk Tagger with Context-Dependent Lexicon. *Proceedings of the Joint Conference on Empirical Methods on Natural Language Processing and Very Large Corpus (EMNLP/ VLC'2000)*. Hong Kong.

Zhou GuoDong & Su Jian. (2002). Named Entity Recognition Using a HMM-based Chunk Tagger, *Proceedings of the Conference on Annual Meeting for Computational Linguistics (ACL'2002).* 473-480, Philadelphia.