Available online at www.sciencedirect.com
SCIENCE DIRECT®
Pattern Recognition Letters 26 (2005) 545–553
Pattern Recognition Letters
www.elsevier.com/locate/patrec

Direct modelling of output context dependence in discriminative hidden Markov model

GuoDong Zhou *

Institute for Infocomm Research, Media Semantics, 21 Heng Mui Keng Terrace, Singapore 119613, Singapore

Received 12 August 2003
Available online 18 October 2004

Abstract

This paper proposes a discriminative HMM to directly model output context dependence. The discriminative HMM assumes mutual information independence in its output model that a "hidden" state is only dependent on the outputs and independent on other "hidden" states. As a result, it overcomes the output context independent assumption in the traditional generative HMM. In addition, a dynamic back-off modelling algorithm using constraint relaxation principle is proposed to resolve the data sparseness problem in the discriminative HMM due to the direct modelling of the output context dependence in its output model. The evaluations on part-of-speech tagging and phrase chunking show that the discriminative HMM can effectively capture the output context dependence through its output context dependent output model and the dynamic back-off modelling algorithm.

Keywords: Discriminative hidden Markov model; Output context dependence; Mutual information independence assumption; Dynamic back-off modelling algorithm; Constraint relaxation principle

1. Introduction

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

In principle, given an output sequence $O_1^n = o_1, o_2, \ldots, o_n$, the goal of HMM is to find a

* Tel.: +65 68745055; fax: +65 67755014.
E-mail address: zhougd@i2r.a-star.edu.sg

0167-8655/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2004.09.009

stochastic optimal state sequence $S_1^n = s_1, s_2, \ldots, s_n$ that maximizes $P(S_1^n|O_1^n)$.

$$S^* = \arg\max_{S_1^n} \log P(S_1^n|O_1^n) \quad (1)$$

By applying Baye's rule, Eq. (1) can be rewritten as

$$\begin{aligned} S^* &= \arg\max_{S_1^n} \{\log P(S_1^n|O_1^n)\} \\ &= \arg\max_{S_1^n} \{\log P(S_1^n) + \log P(O_1^n|S_1^n) - \log P(O_1^n)\} \\ &= \arg\max_{S_1^n} \{\log P(S_1^n) + \log P(O_1^n|S_1^n)\} \end{aligned} \quad (2)$$

The first term $\log P(S_1^n)$ in Eq. (2) is called the state transition model while the second term $\log P(O_1^n|S_1^n)$ is called the output model. In literature, an output context independent assumption is made in the output model: successive outputs are independent given the state sequence (Rabiner, 1989). That is,

$$P(O_1^n|S_1^n) = \prod_{i=1}^{n} P(o_i|S_1^n) \quad (3)$$

By applying the assumption (3), Eq. (2) can be rewritten as

$$S^* = \arg\max_{S_1^n} \left\{ \log P(S_1^n) + \sum_{i=1}^{n} \log P(o_i|S_1^n) \right\} \quad (4)$$

Therefore, the above HMM consists of two models: the state transition model $\log P(S_1^n)$ and the output model $\sum_{i=1}^{n} \log P(o_i|S_1^n)$. This HMM can be called generative HMM because $P(o_i|S_1^n)$ in the output model directly builds the "hidden" states-conditional probability distribution over the outputs. $P(S_1^n)$ in the state transition model can be computed by applying chain rules. For example, in ngram modelling (Jelinek, 1989; Katz, 1987), each state is assumed to be probabilistically dependent on the $N-1$ previous states. Normally, the Viterbi algorithm (Viterbi, 1967) is implemented to find the most likely state sequence based on the state transition model and the output model.

The major limitation of the generative HMM is the output context independent assumption in its output model that successive outputs are independent given the state sequence. While the dependence between successive states can be well modeled by its state transition model using ngram moldeing, the generative HMM fails to directly capture the output context dependence although some of output context dependence can be indirectly captured by its state transition model. In this way, the generative HMM can be also called output context independent HMM.

This paper proposes a discriminative HMM to directly model output context dependence. It is discriminative in that it directly models the "hidden" states given successive outputs. The discriminative HMM assumes mutual information independence in its output model that a "hidden" state is only dependent on the outputs and independent on other "hidden" states. As a result, it overcomes the output context independent assumption in the traditional generative HMM. With the same state transition model as in the generative HMM, it directly captures the output context dependence through an output context dependent output model. In this way, the discriminative HMM can be also called output context dependent HMM. In addition, a dynamic back-off modelling algorithm using constraint relaxation principle is proposed to resolve the data sparseness problem in the discriminative HMM due to the direct modelling of the output context dependence in its output model. The evaluations on part-of-speech tagging and phrase chunking show that the discriminative HMM can effectively capture the output context dependence through its output context dependent output model and the dynamic back-off modelling algorithm.

The layout of this paper is as follows. In Section 2, we will propose the discriminative HMM and explain how it can directly model the output context dependence in its output model while Section 3 presents the constraint relaxation principle and the dynamic back-off modelling algorithm to resolve the data sparseness problem in the output model of the discriminative HMM. Then Section 4 introduces their applications in POS tagging and phrase chunking while experimental results are given in Section 5. Finally, some conclusion will be provided in Section 6.

2. Discriminative hidden Markov model

Instead of applying Baye's rule, we can rewrite Eq. (1) as

$$S^* = \arg\max_{S_1^n}\{\log P(S_1^n|O_1^n)\} = \arg\max_{S_1^n}\left\{\log P(S_1^n) + \log\frac{P(S_1^n, O_1^n)}{P(S_1^n)\cdot P(O_1^n)}\right\} \quad (5)$$

The first term $\log P(S_1^n)$ in Eq. (5) is called the state transition model while the second term $\log\frac{P(S_1^n,O_1^n)}{P(S_1^n)\cdot P(O_1^n)}$ is called the output model. Obviously, the output model captures the mutual information between the state sequence S_1^n and the output sequence O_1^n. Here, we have a mutual information independence assumption in the output model

$$MI(S_1^n, O_1^n) = \sum_{i=1}^{n} MI(s_i, O_1^n)$$

or

$$\log\frac{P(S_1^n, O_1^n)}{P(S_1^n)\cdot P(O_1^n)} = \sum_{i=1}^{n}\log\frac{P(s_i, O_1^n)}{P(s_i)\cdot P(O_1^n)} \quad (6)$$

That is, an individual state is only dependent on the output 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 already been captured by the state transition model $\log P(S_1^n)$.

By applying assumption (6) into Eq. (5), we have

$$S^* = \arg\max_{S_1^n}\left\{\log P(S_1^n) + \sum_{i=1}^{n}\log P(s_i|O_1^n) - \sum_{i=1}^{n}\log P(s_i)\right\} \quad (7)$$

The above HMM also consists of two models: the same state transition model $\log P(S_1^n)$ as in the generative HMM and the output model $\sum_{i=1}^{n}\log P(s_i|O_1^n) - \sum_{i=1}^{n}\log P(s_i)$. Here, $\log P(s_i)$ in the output model cancels prior state probability from $\log P(s_i|O_1^n)$ in the output model. This HMM can be called discriminative HMM because $P(s_i|O_1^n)$ in the output model directly models the "hidden" states given successive outputs. Given the same state transition model for both the generative and discriminative HMMs, the difference between them lies in their output models in that the output model of the discriminative HMM directly captures the output context dependence between successive outputs in determining the "hidden" states while the output model of the generative HMM fails to do so. That is, the output model of the discriminative HMM overcomes the output context independent assumption in the generative HMM and becomes output context dependent. In this way, the discriminative HMM can also be called output context dependent HMM.

Compared with the generative HMM, the problem with the discriminative HMM lies in the data sparseness problem raised by $\sum_{i=1}^{n}\log P(s_i|O_1^n)$ in its output model, which tries to capture the output context dependence in determining the "hidden" states. 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. Generally there are two existing approaches to resolve this problem: linear interpolation (Jelinek, 1989) and back-off (Katz, 1987). Although back-off is generally inferior to linear interpolation and has another problem in that it exhibits a discontinuity around the point where the back-off decision is made, the back-off approach is simple, compact and often almost as good as linear interpolation. However, these two approaches only work well when the number of different information sources is limited. When long context is considered, the number of different information sources is exponential. In next section, we propose a constraint relaxation principle and a dynamic back-off modelling algorithm to resolve the data sparseness problem in the output model of the discriminative HMM where long context is considered.

3. Dynamic back-off modelling algorithm

The main challenge for the discriminative HMM is to reliably estimate $P(s_i|O_1^n)$ in its output model. For efficiency, we can always assume

$P(s_i|O_1^n) \approx P(s_i|E_i)$, where the pattern entry $E_i = o_{i-N} \cdots o_i \cdots o_{i+N}$. That is, we only consider the output context dependence in a window of $2N+1$ outputs (e.g. we only consider current output, previous output and next output when $N=1$). For convenience, we denote $P(\bullet|E_i)$ as the probability distribution among the "hidden" states related with the pattern entry E_i and $P(s_i|E_i)$ as the probability of an individual state s_i related with E_i.

Here, we recast reliably estimating $P(\bullet|E_i)$ as a problem of finding an optimal frequently occurring pattern entry E_i^0, which should occur at least N (e.g. 10) times in the training corpus and satisfy $P(\bullet|E_i) \approx P(\bullet|E_i^0)$. To do so, all the frequently occurring pattern entries are extracted from the training corpus and stored in a pattern entry dictionary *FrequentEntryDictionary*. In order to further limit the size of *FrequentEntryDictionary* besides the window factor, we also define a valid set of pattern entry forms *ValidEntryForm* to consider only the most informative information sources. Generally, *ValidEntryForm* can be determined according to the applications. In Section 4, we will give two examples.

In this paper, a dynamic back-off modelling algorithm using constraint relaxation principle is proposed to estimate $P(\bullet|E_i)$. Normally, an output contains several features and the constraints include all the features in the outputs of E_i. With the large number of ways in which the constraints could be relaxed, the challenge is how to avoid intractability and drop in efficiency. Two restrictions are applied to keep the relaxation process tractable and manageable

- Relaxation is done through iteratively dropping a constraint from the pattern entry.
- The pattern entry after relaxation should also have a valid form as defined in *ValidEntryForm*.

By means of constraint relaxation principle, the dynamic back-off modelling algorithm estimates $P(\bullet|E_i)$ as follows:

Dynamic back-off modelling algorithm for computing $P(\bullet|E_i)$

Assume an initial pattern entry $E_i = o_{i-2}o_{i-1}o_io_{i+1}o_{i+2}$.
Assume a valid set of pattern entry forms *ValidEntryForm*
Assume *FrequentEntryDictionary* is the pattern entry dictionary which stores all the frequently occurring pattern entries in the training corpus with the state probability distributions.
Assume *likelihood*(E_i) the likelihood of a pattern entry E_i as the optimal frequently occurring pattern entry to be found.
Assume E_i^0 is the optimal frequently occurring pattern entry to be found.

BEGIN-of-algorithm
IF $E_i \in$ *FrequentEntryDictionary*
THEN
BEGIN
  Set $E_i^0 = E_i$
  RETURN $P(\bullet|E_i) = P(\bullet|E_i^0)$
END
Compute *likelihood*(E_i)
Initialize *InputEntrySet* = $\{\langle E_i, likelihood(E_i)\rangle\}$ and *OutputEntrySet* = {}
LOOP
  DO for every entry$\langle E_j, likelihood(E_j)\rangle \in$ *InputEntrySet*
    DO for every constraint in E_j
      Assume C_j^k is the constraint in E_j
      Set $E_j\prime = E_j - C_j^k$ as the pattern entry after relaxation of the constraint C_j^k in E_j
      IF E_j' conforms to *ValidEntryForm*
      THEN
      BEGIN
        Compute *likelihood*(E_j')
        Set *OutputEntrySet* = *OutputEntrySet* + $\{\langle E_j', likelihood(E_j')\rangle\}$
      END IF
    END DO
  END DO
  Get $E_i^0 = \underset{\substack{\langle E, likelihood(E)\rangle \in OutputEntrySet \\ E \in FrequentEntryDictionary}}{\arg\max} likelihood(E)$
  IF E_i^0 is not found
  THEN Set *InputEntrySet* = *OutputEntrySet* and *OutputEntrySet* = {}
  ELSE exit LOOP
END LOOP
RETURN $P(\bullet|E_i) = P(\bullet|E_i^0)$
END-of-algorithm

The dynamic back-off modelling algorithm solves the problem by iteratively relaxing a constraint in the initial pattern entry E_i until a near optimal frequently occurring pattern entry E_i^0 is reached. If E_i frequently occurs, we just return E_i as E_i^0. Otherwise, a near optimal entry E_i^0 will be found by iteratively relaxing the initial entry E_i. At every loop, we have a set of entries *InputEntrySet* as input and return a set of entries *OutputEntrySet* as output, which is the input for the next loop. The entries in *InputEntrySet* and *OutputEntrySet* are ranked according to their likelihoods as the optimal frequently occurring pattern entry to be found. Here, the likelihood of a pattern entry is determined by the number and the positioning of its non-empty constraints according to the following two principles:

- The more number of non-empty constraints in a pattern entry, the more informative the pattern entry is.
- The closer a non-empty constraint is to current output, the more informative the non-empty constraint is.

InputEntrySet is initialized as $\{\langle E_i, likelihood(E_i)\rangle\}$ and *OutputEntrySet* can be generated by relaxing any constraint (validated by *ValidEntryForm*) in every pattern entry of *InputEntrySet*. If no pattern entry in *OutputEntrySet* frequently occurs, the pattern entries in *OutputEntrySet* is feed back to *InputEntrySet* for next loop and a further set of pattern entries *OutputEntrySet* can be generated by relaxing any constraint in each pattern entry of *InputEntrySet*. The process continues until E_i^0 is found. In order to remain efficient, only top N (e.g. 5) pattern entries are kept in *InputEntrySet* and *OutputEntrySet* according to their likelihoods.

4. Part-of-speech tagging and phrase chunking

In order to evaluate the discriminative HMM and the dynamic back-off modelling algorithm, we have applied them in the applications of part-of-speech tagging and phrase chunking.

4.1. Part-of-speech tagging

For part-of-speech (POS) tagging, 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 word formation pattern sequence, while the "hidden" states are POS tags. Here, the word formation pattern is used to deal with the unknown words and consists of two parts:

- Suffixes: Here, we include top 200 most frequently occurring suffixes, such as ∼ing, ∼tion, ∼ed and ∼ies.
- Capitalization and digitalization: Usually, capitalization and digitalization information can be used to differentiate between entity nouns and others. Table 1 shows a complete list of word formation patterns with the descending order of priority.

4.2. Phrase chunking

For phrase chunking, 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 and Brants (1998) and Zhou and Su (2002). Here, a structural tag consists of three parts:

Table 1
F_{WFP}: word formation patterns

Pattern (1–9)	Example	Pattern (10–17)	Example
Comma	,	CapMixAlpha	IgM
Dot	.	LowMixAlpha	kDa
Parenthesis	( ) [ ] {}	AlphaDigitAlpha	H2A
RomanDigit	II	AlphaDigit	T4
GreekLetter	Beta	DigitAlphaDigit	6C2
Number	1.25	DigitAlpha	19D
OneCap	T	Lowcase	Will
AllCaps	CSF	Others	$
CapLowAlpha	All		

- *Boundary Category (BOUNDARY)*. It is a set of four values: "O"/"B"/"M"/"E", where "O" means that current word is a whole phrase and "B"/"M"/"E" means that current word is at the beginning/in the middle/at the end of an 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, POS is added into the structural tag to represent more accurate state transition model and output model.

For example, given following POS tagged sentence as the output 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 decoded sequence of "hidden" structural tags:

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 phrase chunking 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./.]

5. Experimental results

The corpus used in part-of-speech (POS) tagging is the PENN TreeBank (Marcus et al., 1993) of one million words (25 sections) while the corpus used in phrase chunking is extracted from the same PENN Treebank by a program provided by Sabine Buchholz from Tilbug University. All the evaluations are five-fold cross-validated. For POS tagging, we only use accuracy to measure the performance. For phrase chunking, we use precision, recall and *F*-measure. Here, accuracy is the percentage of the words that are labeled with correct POS tags; precision (P) is the percentage of predicted phrase chunks that are actually correct; the recall (R) is the percentage of correct phrase chunks that are actually found and *F*-measure is the weighted harmonic mean of precision and recall: $F = \frac{(\beta^2+1)RP}{\beta^2 R+P}$ with $\beta^2 = 1$ (Rijsbergen, 1979).

For each application using the discriminative HMM with the dynamic back-off modelling algorithm, two settings are applied:

- Output context independent setting: the valid set of pattern entry forms *ValidEntryForm* in the dynamic back-off modelling algorithm carries no output context information in the output model. That is, $ValidEntryForm = p_i\{w_i\} = \{p_i, p_i w_i\}$. $p_i\{w_i\}$ means compulsory p_i with optional w_i.
- Output context dependent setting: the valid set of pattern entry forms *ValidEntryForm* in the dynamic back-off modelling algorithm carries various output context information sources in a window of five words. Here, $ValidEntryForm = \{\{p_{i-2}\{w_{i-2}\}\}p_{i-1}\{w_{i-1}\}\}p_i\{w_i\}\{p_{i+1}\{w_{i+1}\}\ \{p_{i+2}\{w_{i+2}\}\}\}$.

For comparison, we also give the performance of the generative HMM for each application, where a window of five states is considered in its output model. Here, back-off 5 gram modelling (Katz, 1987) is used to resolve the data sparseness problem in the output model of the generative HMM. Besides, back-off trigram modelling (Katz, 1987) is applied to resolve the data sparseness problem in the state transition models of both the generative and discriminative HMMs.

Table 2 shows that the generative HMM, the discriminative HMMs with independent and dependent output contexts achieve accuracies of 96.24%, 96.18% and 96.93%, respectively for POS tagging. It shows that the discriminative HMM with independent output context is compa-

Table 2
Comparison of performance between the generative HMM and the discriminative HMMs with independent and dependent contexts on POS tagging

POS tagging	Accuracy (%)
Generative HMM	96.34
Discriminative HMM with independent output context	96.28
Discriminative HMM with dependent output context	96.93

rable to the generative HMM for POS tagging. It also shows that the discriminative HMM with dependent output context outperforms the generative HMM by reducing 16% of errors for POS tagging. This means that, given the same state transition model as in the generative HMM, the output model of the discriminative HMM and the dynamic back-off modelling algorithm can effectively capture more output context dependence than the output model of the generative HMM in determining POS tags.

Table 3 shows that the generative HMM, the discriminative HMM with independent and dependent output contexts achieve *F*-measures of 91.48, 91.25 and 94.14, respectively for phrase chunking. It shows that the discriminative HMM with independent output context is comparable to the generative HMM for phrase chunking. It also shows that the discriminative HMM with dependent output context significantly outperforms the generative HMM by reducing 28% of errors for phrase chunking. This means that, given the same state transition model as in the generative HMM, the output model of the discriminative HMM and the dynamic back-off modelling algorithm can effectively capture much more output context dependence than the output model of the generative HMM in determining phrase chunks.

Table 3
Comparison of performance between the generative HMM and the discriminative HMMs with independent and dependent contexts on phrase chunking

Phrase Chunking	Precision (%)	Recall (%)	*F*-measure
Generative HMM	90.72	92.25	91.48
Discriminative HMM with independent output context	90.38	92.14	91.25
Discriminative HMM with dependent output context	93.71	94.03	93.87

One important question is the extra overhead of the discriminative HMM (with dependent output context) over the generative HMM. The extra overhead comes from two aspects. The first is the extra memory requirement due to the output model. Here, 43% more memory is used to store 43% extra pattern entries in the output model of the discriminative HMM (here, the threshold is set as 10 for frequently occurring pattern entries) than that of the generative HMM. This accounts for 21% and 15% of the whole memory overhead for POS tagging and phrase chunking, respectively. The second is the extra CPU overhead due to the dynamic back-off modelling algorithm. We find that the dynamic back-off modelling algorithm occupies 14% and 23% of the whole CPU overhead for POS tagging and phrase chunking, respectively. We think the extra overhead of the discriminative HMM over the generative HMM is acceptable, considering the performance gained for these two applications.

Another important question is about the effect of different training data sizes on the performance of the discriminative HMM over the generative HMM. Fig. 1 answers the question. It shows the discriminative HMM increasingly outperforms the generative HMM for POS tagging and phrase chunking as the training data increases. This means that the discriminative HMM is much more data-driven than the generative HMM due to the direct modelling of the output context dependence in the output model of the discriminative HMM. It also shows that the contribution of the discriminative HMM over the generative HMM is increasingly more significant for phrase chunking than for POS tagging as the training data increases. The major reason may be that phrase chunking is much more output context dependent than POS tagging. Another reason is the ability of the discriminative HMM in dealing with output context dependence.

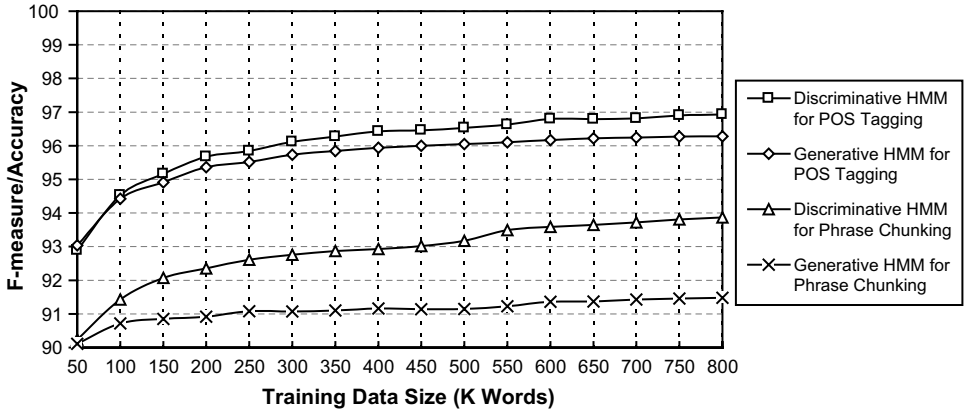

Fig. 1. The effect of different training data sizes on the performance of POS tagging and phrase chunking for the discriminative HMM and the generative HMM.

6. Conclusion

In this paper, we first introduce the generative HMM. The major limitation of the generative HMM is the output context independent assumption in its output model that successive outputs are independent given the state sequence. While the dependence between successive states can be well modelled by its state transition model using ngram moldeing, the generative HMM fails to directly capture the output context dependence although some of output context dependence can be indirectly captured by its state transition model. In this way, the generative HMM can be also called output context independent HMM. Then this paper proposes a discriminative HMM, which directly models the ''hidden'' states given successive outputs through a mutual information independence assumption in its output model that a ''hidden'' state is only dependent on the outputs and independent on other ''hidden'' states. With the same state transition model as in the generative HMM, it directly captures the output context dependence through an output context dependent output model and overcomes the output context independent assumption in the generative HMM. In this way, the discriminative HMM can be also called output context dependent HMM. Finally, a dynamic back-off modelling algorithm using the constraint relaxation principle is proposed to resolve the data sparseness problem in the discriminative HMM due to the direct modelling of the output context dependence in its output model. The evaluations show that the discriminative HMM with the dynamic back-off modelling algorithm performs significantly better than the generative HMM by reducing 16% and 28% of errors on POS tagging and phrase chunking, respectively. Therefore, it can be concluded that the discriminative HMM can effectively capture the output context dependence through its output context dependent output model and the dynamic back-off modelling algorithm.

References

Bikel, D.M., Schwartz, R., Weischedel, R.M., 1999. An algorithm that learns what's in a name. Mach. Learn. 34, 211–231 (Special issue on NLP).

Brants, T., Skut, W., Krenn, B., 1997. Tagging Grammatical Functions. In: Procceedings of the Conference on Empirical Methods on Natural Language Processing (EMNLP'1997). Brown University, RI.

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

Jelinek, F., 1989. Self-Organized Language Modeling for Speech Recognition. In: Waibel, A., Lee, K.-F. (Eds.), Readings in Speech Recognition. Morgan Kaufmann, pp. 450–506.

Katz, S.M., 1987. Estimation of probabilities from sparse data for the language model component of a speech recognizer. IEEE Trans. Acoust. Speech Signal Process. 35, 400–401.

Marcus, M., Santorini, B., Marcinkiewicz, M.A., 1993. Bulding a large annotated corpus of English: The Penn Treebank. Comput. Linguist. 19 (2), 313–330.

Merialdo, B., 1994. Tagging english text with a probabilistic model. Comput. Linguist. 20 (2), 155–171.

Rabiner, L.R., 1989. A tutorial on hidden Markov models and selected applications in speech recognition. Proceed. IEEE 77 (2), 257–286.

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

Skut, W., Brants, T., 1998. Chunk Tagger—Statistical Recognition of Noun Phrases. In: Proceedings of the ESSLLI'98 workshop on Automatic Acquisition of Syntax and Parsing. University 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 Trans. Infor. Theory IT (13), 260–269.

Weischedel, R., Meteer, M., Schwartz, R., Ramshaw, L., Palmucci, J., 1993. Coping with ambiguity and unknown words through probabilistic methods. Comput. Linguist. 19 (2), 359–382.

Zhou GuoDong, Su Jian, 2000. Error-driven HMM-based chunk tagger with context-dependent lexicon. In: 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. In: Proceedings of the Conference on Annual Meeting for Computational Linguistics (ACL'2002). Philadelphia. pp. 473–480.