International Symposium on Chinese Spoken Language Processing, pp. 141-144, Oct. 13-15, 2000, Beijing

WORD-CLASS STOCHASTIC MODEL IN A SPOKEN LANGUAGE DIALOGUE SYSTEM

YAN Pengju, ZHENG Fang, XU Mingxing, HHUANG Yinfei
Center of Speech Technology, State Key Laboratory of Intelligent Technology and Systems,
Department of Computer Science & Technology, Tsinghua University, Beijing
{*yanpj,fzheng,xumx,huangyf*}*@sp.cs.tsinghua.edu.cn*

ABSTRACT

Spoken Language Understanding (SLU) is a key component of spoken dialogue systems. One popular SLU method is to use the continuous speech recognizer where the Part-Of-Speech (POS) tagging is employed to determine the underlying word-class sequences. We present here a Word-Class Stochastic Model (WCSM) to describe the temporal word/word-class sequences, which is fit into the standard paradigm of the Hidden Markov Models (HMMs). The model training is done on the basis of a general-purposed, large-vocabulary-sized, labeled corpus, which makes the model comparatively easy to construct. We apply the model to a prototype dialogue system named *EasyNav*, and the use of domain-specific knowledge, i.e., semantic-meaningful keywords, helps to increase the speed and accuracy of the POS tagging process.

1. INTRODUCTION

The spoken language understanding (analysis/parsing) (SLU) is one of the most significant parts in spoken language understanding and dialogue systems. The performance of SLU component can greatly affect the performance of spoken dialogue systems. Currently, there exist two popular SLU approaches. One is based on the continuous speech recognition where the run-through utterances are fully recognized for future use. The other one uses the keyword/concept based technology where only meaningful and promising acoustic results are considered. In the former case, conventional word stochastic models (WSM) or novel word-class stochastic models (WCSM) are employed to determine the word/word-class sequence given the syllable lattice. Compare with the WSM, WCSM achieves higher performance and lower time complexity.

Mathematicians always try to find systematic and elegant ways to characterize real-word processes. Markov models (MMs) and hidden Markov models are introduced in the late 1960s or early 1970s, which are very rich in mathematical structure and hence can form the theoretical basis for use in a wide range of applications [1]. Speech signals are so complicated that there are still no exact physical models available to effectively characterize them so far. However, while adopting HMMs, speech signals can be simplified to follow the pattern that the transformation of some hidden states goes under the transformation of some other surface/visible states. Like other two-level hierarchical processes, the temporal transformation process of the word/word-class sequences can be described by HMMs because of its "hidden-surface" nature. Now MMs and HMMs are prevalently used in speech processing and stochastic language processing. Our WCSM, for use in POS tagging, is fit into the standard paradigm of HMMs and show the advantage of easily training.

For use in a real-world system, the assistance or constraints of domain-specific knowledge can always helps a certain general-purposed model to improve its performance, so does to our WCSM. Several keyword-classes and hundreds of keywords are used in our model that apparently increases the speed and accuracy of the POS tagging process.

EasyNav, designed by the Center of Speech Technology of Tsinghua University, is currently a text-based dialogue system aiming at providing users with Tsinghua University campus navigation information. Syntactic-rule-based-parsing demands for POS tagging and WCSM is incorporated to do that.

The model training is based on a labeled general-purposed corpus and we use conventional Viterbi decoding algorithm to determine the most possible word-class sequence. Experiments on *EasyNav* show satisfying performances of the WCSM.

In the following sections we will report on the efforts and experiences of designing and applying of the WCSM on *EasyNav*. The paper is structured as follows. Section 2 introduces the architecture of *EasyNav*; Section 3 gives the detailed description of our model; Section 4 presents the model training and POS tagging algorithm; Section 5 gives the applying results; and conclusions are given in section 6.

2. *EASYNAV* OVERVIEW

2.1 *EasyNav* Architecture

EasyNav, our first prototype effort, is currently a text-based dialogue system the purpose of which is to provide users with Tsinghua University campus information on commerce, geography and official businesses. To some extent, it serves as a platform for us to do some research experiments, including language understanding and knowledge representation technologies.

Figure 1 depicts the system architecture.

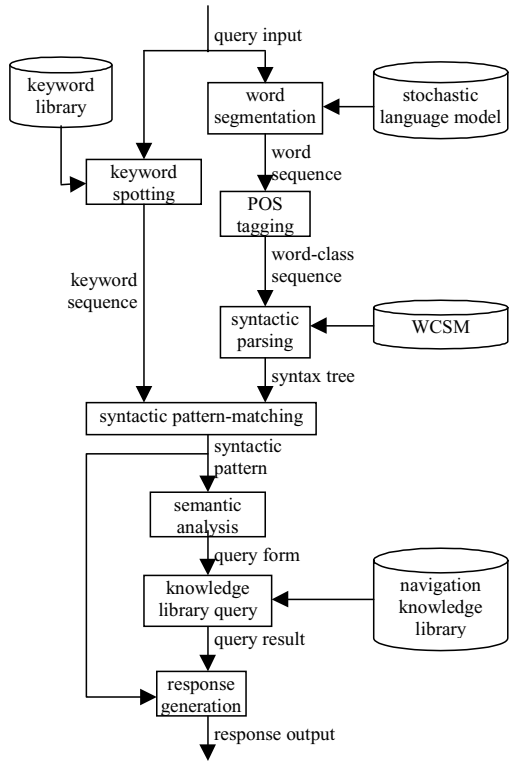

Figure 1. EasyNav's modules and data flowing

2.2 System Strategies

EasyNav uses a syntactic parsing and a syntactic pattern-matching scheme for language understanding. It currently follows the single-query-turn working modality and its input and output are both written texts.

One of the two key features of *EasyNav* is that we use a hybrid frame representation method to model the domain-specific knowledge. The model frame, referred to as a data-independent multi-branch tree hierarchical structure, has a strong expressing ability and hence can be easily expanded and instantiated onto new applications.

The other feature is that we make use of the traditional two-stage understanding technique, including the syntactics and the semantics, and the detailed strategies can be described as follows. The word segmentation employs an n-gram language model to segment the user's sentence into word sequence, while the POS tagging determines the underlying word-class sequence. Syntactic-rule-based parser checks the validity of the sentence and the resultant syntax tree plus the keyword information are submitted to the matching component for producing the syntactic pattern. Afterwards, a query form is assembled to the semantic analyzer as the query request. The query module searches in the knowledge base, and the results as well as the syntactic pattern helps generating the ultimate response output to the user.

The incorporated WCSM is the basis of the syntactic rule based semantic parsing and spoken language understanding. The use of WCSM has an inherent ability to label a never-seen-before sentence because the probabilities of word-class sequences rather than those of word sequences are considered in it.

3. WORD-CLASS STOCHASTIC MODEL

Hidden Markov Model is the footstone of our word-class stochastic model. In this section, we will first introduce the standard (discrete) Hidden Markov Model before describing our WCSM.

3.1 MMs and HMMs Routines

Definition Ⅰ. A stochastic sequence $\{X_n, n \geq 0\}$ is called a Markov chain on the time space $T = \{0,1,2,\cdots\}$ and the state space $S = \{1,2,\cdots\}$, if for any $s_0, s_1, \ldots, s_n, s_{n+1} \in S, n \in T$ and $P(X_0 = s_0, X_1 = s_1, \cdots, X_n = s_n) > 0$ it holds that

$$P(X_{n+1} = s_{n+1} \mid X_0 = s_0, X_1 = s_1, \cdots, X_n = s_n) = P(X_{n+1} = s_{n+1} \mid X_n = s_n) \quad (1)$$

Furthermore we consider only those processes in which the right-hand side of (1) is independent of time, i.e.,

$$\forall i, j \in S, \exists p_{ij} \in R, \forall n \in T : P\{X_{n+1} = j \mid X_n = i\} = p_{ij} . \quad (2)$$

We call $\mathbf{P} = \{p_{ij}\}$ as the state transition probability matrix and $\boldsymbol{\pi} = (\alpha_0 = P(s_0), \alpha_1 = P(s_1), \cdots)$ as the initial distribution vector of $\{X_n\}$. □

The key point of Markov chain is that if given the "now", the "future" is independent of the "past". This hypothesis is too strict, but it makes the problem easy to solve.

Definition Ⅱ. A stochastic vector sequence $\{(X_n, Y_n), n \geq 0\}$ is called a hidden Markov chain on the time space $T = \{0,1,2,\cdots\}$, the inner state space $S = \{1,2,\cdots\}$ and the output state space $O = \{1,2,\cdots\}$, if $\{X_n, n \geq 0\}$ is a Markov chain on T and S, and its state transition probability matrix and initial distribution vector are $\mathbf{A} = \{a_{ij}\}$ and $\boldsymbol{\pi}$ respectively, and for any

$s_0, s_1, \ldots, s_n = i \in S, o_n = j \in O, n \in T$ and $P(X_0 = s_0, X_1 = s_1, \cdots, X_n = s_n) > 0$ it holds that

$$P(Y_n = o_n \mid X_0 = s_0, X_1 = s_1, \cdots, X_n = s_n) = P(Y_n = o_n \mid X_n = s_n) = b_{ij}. \quad (3)$$

We call $\mathbf{A}$, $\mathbf{B} = \{b_{ij}\}$ and $\boldsymbol{\pi}$ as the state transition probability matrix, output probability matrix and initial distribution vector of $\{(X_n, Y_n)\}$ respectively. □

There is an implicit hypothesis in each definition that the current state only depends on the past state(s), since we know that real-world processes follow that rule.

3.2 Model Description

The occurrence probability of sentence $W_1^N = (w_1, w_2, \ldots, w_N)$ can be estimated as

$$P(W_1^N) = P(w_1)\prod_{n=2}^{N} P(w_n \mid W_1^{n-1}) = P(w_1)\prod_{n=2}^{N} P(w_n \mid W_{n-m+1}^{n-1}), \quad (4)$$

where we assume that the sentences follow the Markov chain hypothesis, i.e., the current word depends only on the previous $m-1$ words.

The grammars where $m = 1,2,3$ are called the uni-gram, bi-gram and tri-gram respectively. Experiences from speech experts state that bi-gram or tri-gram is sufficient for speech processing.

Furthermore we take the process of word/word-class changing with time as a hidden Markov chain, words vs. output states and word-classes vs. inner states. In other words, we assume that the current word-class depends only on the immediate previous word-class, and current word depends only on the current word-class.

The co-occurrence probability of a word w and its corresponding class c can be written as

$$P(w, c) = P(c)P(w \mid c), \quad (5)$$

where $P(c)$ is called the **word-class probability** and $P(w \mid c)$ is the **word-attribute probability**. Moreover, the co-occurrence probability of a word sequence and the corresponding word-class sequence can be estimated as

$$\begin{aligned} P(W_1^N, C_1^N) &= P(w_1, c_1)\prod_{n=2}^{N} P(w_n, c_n \mid W_1^{n-1}, C_1^{n-1}) \\ &= P(c_1)P(w_1 \mid c_1)\prod_{n=2}^{N} P(c_n \mid W_1^{n-1}, C_1^{n-1})P(w_n \mid W_1^{n-1}, C_1^{n}), \\ &= P(c_1)P(w_1 \mid c_1)\prod_{n=2}^{N} P(c_n \mid C_{n-m+1}^{n-1})P(w_n \mid c_n) \end{aligned} \quad (6)$$

where $P(c_n \mid C_{n-m+1}^{n-1})$ corresponds to the state transition probability in definition Ⅱ and $P(w_n \mid C_n)$ the output probability.

As we can see, word-class probability (word-class n-gram) and word-attribute probability make up the whole model parameters.

4. APPLYING TO *EASYNAV*

4.1 Model Training

In the absence of adequate domain-specific corpus, we make use of a general-purposed labeled corpus instead, in which the lexicon size is 50,624 and the size of the word-class list is 91[2]. The form of a labeled item is as

$$[(A, B, C), (R, S, T)]: \quad c(A, B, C), \quad (7)$$

where C/T means the current word/word-class pair, B/S means the previous pair and A/R means the pair of *time-before-previous*. $[(A, B, C), (R, S, T)]$ indicates that the word-class-triple appearing most frequently for word-triple (A, B, C) in the corpus is (R, S, T), while $c(A, B, C)$ denotes the occurrence time of (A, B, C) in the corpus.

The training (estimate) formulas can be written as follows,

$$\begin{cases} C(R,S,T) = \sum_{A,B,C:[(A,B,C),(R,S,T)]} count(A,B,C) \\ C(R,S) = \sum_{A,B,C,T:[(A,B,C),(R,S,T)]} count(A,B,C) \\ C(R) = \sum_{A,B,C,S,T:[(A,B,C),(R,S,T)]} count(A,B,C) \\ C(A,R) = \sum_{B,C,S,T:[(A,B,C),(R,S,T)]} count(A,B,C) \end{cases}, \quad (8)$$

$$\begin{cases} P(T \mid R,S) = C(R,S,T)/C(R,S) \\ P(S \mid R) = C(R,S)/C(R) \\ P(R) = C(R)\Big/\sum_{R_i} C(R_i) \\ P(A \mid R) = C(A,R)/C(R) \end{cases}, \quad (9)$$

where $C(\bullet)$ and $P(\bullet)$ stands for the frequency and distribution function respectively. It can be seen that these formulas meet the normalization requirements.

Compare with the standard HMM training methods, the proposed model needs much smaller training data and is much simpler. Since the model is used in a fixed domain and the number of real concerned words is comparatively small, the data-sparseness is not remarkable.

4.2 Design and Implementation of POS Tagging Algorithm

The objective of POS tagging is to assign each word with a most likely word-class. There exist two kinds of tagging criteria,

$$\tilde{C}_1^N = \left(\tilde{c}_1, \cdots, \tilde{c}_N : \tilde{c}_n = \arg\max_{c_n} P\left(c_n \mid W_1^N\right), \quad 1 \le n \le N \right), \quad (10)$$

and

$$\tilde{C}_1^N = \arg\max_{C_1^N} P\left(C_1^N \mid W_1^N\right). \quad (11)$$

Equation (11) takes the sentence as a whole thus is more reasonable in our application. Assume the word sequence is given, we can obtain from Equation (11) that

$$\begin{aligned} \tilde{C}_1^N &= \arg\max_{C_1^N} P\left(C_1^N \mid W_1^N\right) \\ &= \arg\max_{C_1^N} P\left(W_1^N, C_1^N\right) / P\left(W_1^N\right) \\ &= \arg\max_{C_1^N} P\left(W_1^N, C_1^N\right) \end{aligned} . \quad (12)$$

Furthermore if we apply the hypothesis as in Equation (6) we obtain

$$\tilde{C}_1^N = \arg\max_{C_1^N} \alpha(c_1) p_b(w_1 \mid c_1) \prod_{n=2}^{N} \left[p_a(c_n \mid c_{n-1}) p_b(w_n \mid c_n) \right], \quad (13)$$

where substitutions of $m = 2$, $\alpha(c_1) = P(c_1)$, $p_a(c_n \mid c_{n-1}) = P(c_n \mid c_{n-1})$, and $p_b(w_n \mid c_n) = P(w_n \mid c_n)$ are taken.

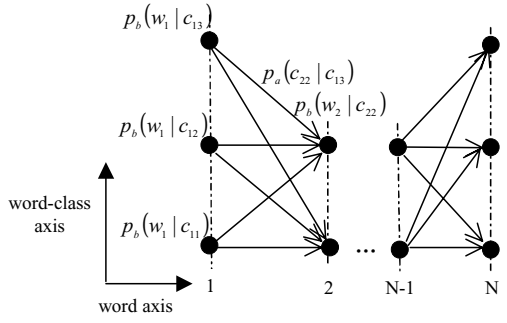

Figure 2. Search lattice

Hence the POS tagging can be regarded as a search process in the word/word-class lattice, as illustrated in Figure 2. In the figure w_n denotes the word at time n, while node $c_{n,i}$ the i-th possible word-class of w_n. The decoding algorithm can be described as follows:

$$\begin{cases} s_{1i} = \alpha(c_{1i}) p_b(w_1 \mid c_{1i}), \quad i = 1, \ldots, M_1 \\ s_{n,i} = \max\limits_{1 \le j \le M_{n-1}} \left[s_{n-1,j} p_a(c_{n,i} \mid c_{n-1,j}) p_b(w_n \mid c_{n,i}) \right] \\ \qquad n = 2, \cdots, N; i = 1, \ldots, M_n \\ B_{n,i} = \arg\max\limits_{1 \le j \le M_{n-1}} \left[s_{n-1,j} p_a(c_{n,i} \mid c_{n-1,j}) p_b(w_n \mid c_{n,i}) \right] \\ \qquad n = 2, \cdots, N; i = 1, \ldots, M_n \end{cases}, \quad (14)$$

where s_{ni} denotes the local maximum score at node $c_{n,i}$, $B_{n,i}$ points to the previous node from which the current node take the best score, and M_n indicates the number of word-class types to which w_n may belongs. Equations (15) are used to backtrack the optimal path:

$$\begin{cases} c_N = \arg\max\limits_i s_{Ni} \\ c_n = B_{n+1,c_{n+1}}, \quad n = N-1, \ldots, 1 \end{cases}. \quad (15)$$

Let

$$M = \max_{1 \le n \le N} (M_n), \quad (16)$$

the time perplexity of the algorithm is $O\left(NM^2\right)$.

5. EXPERIMENTAL RESULTS

Being one of the joint components of the modules of word segmentation and syntactic parsing, the running of the POS tagging depends on not only the WCSM but also the system lexicon and user lexicon where the former is mentioned in [2] while the latter is made up by keywords extracted empirically from a corpus collected from our lab members.

The keywords are grouped into a couple of keyword-classes and each keyword-class is assigned a appropriate word-class type. Since the keyword has a unique word-class type and it holds that $N \le 10$ and $M \le 3$, we practically apply full-paths search in *EasyNav* without slowing down the searching speed.

The POS tagging achieves very accurate results and only the top candidate is considered in *EasyNav*. Hundreds of query experiments prove its availability.

6. CONCLUSION

The SLU method using the syntactic parsing can lead to the precise structure results. Our WSCM models the process of word/word-class sequence in the concerned sentence and the model training is very simple on a general-purposed labeled corpus. The POS tagging algorithm gives accurate word-class sequence of the user's sentence and the use of domain-specific knowledge enhances the model performance.

7. REFERENCES

[1] Rabiner L. R., "A tutorial on hidden Markov models and selected applications in speech recognition", *Proceedings of the IEEE*, v 77, n 2, p 257-286.

[2] Zheng F, Song Z.J., Xu M.X, et al. "EasyTalk: A Large-Vocabulary Speaker-Independent Chinese Dictation Machine". *EuroSpeech'99 Proceedings*, Budapest, 1999.