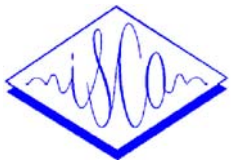

INTERSPEECH 2004 - ICSLP
8th International Conference on Spoken Language Processing
ICC Jeju, Jeju Island, Korea
October 4-8, 2004

Mining of Association Patterns for Language Modeling

Jen-Tzung Chien and Hung-Ying Chen
Department of Computer Science and Information Engineering,
National Cheng Kung University, Tainan, Taiwan, ROC
jtchien@mail.ncku.edu.tw

Abstract

Language modeling using *n*-gram is popular for speech recognition and many other applications. The conventional *n*-gram suffers from the insufficiencies of training data, domain knowledge and long distance language dependencies. This paper presents a new approach to mining long distance word associations and incorporating their mutual information into language models. We aim to discover the associations of *multiple distant words* from training corpus. An efficient algorithm is exploited to merge the frequent word subsets and construct the association patterns. The resulting *association pattern n-gram* is general with a special realization to trigger pair *n*-gram where only associations of two distant words are considered. To improve the modeling, we further compensate the weaknesses of sparse training data via parameter smoothing and domain mismatch via online adaptive learning. The proposed association pattern *n*-gram and several hybrid models are successfully applied for speech recognition. We also find that the incorporation of mutual information of association patterns can significantly reduce the perplexities of language models.

1. Introduction

There is no doubt that the statistical language models using *n*-grams play a decisive role in natural language processing. The applications have been extended to speech recognition, document classification, information retrieval, optical character recognition, machine translation, writing correction, and bio-informatics. The conventional *n*-gram models suffer from three weaknesses: 1) *data sparseness* problem in parameter estimation, 2) *domain mismatch* between training and test corpora and 3) difficulties in modeling *distant word dependencies*. First, the data sparseness is inevitable in *n*-gram modeling because a large number of unseen word combinations exist in training data. It is effective to smooth *n*-gram models so as to prevent zero probabilities in unseen word combinations. A complete survey of smoothing algorithms was conducted in [3]. Witten-Bell smoothing [7] has been widely used for language model smoothing. Secondly, the *n*-gram models are very sensitive to changes in topic on which they were trained. Accordingly, *n*-gram should be adaptive to meet the evolution of new domains. In [5], the general *n*-gram was interpolated by the cache-based *n*-gram to produce the adaptive mixture-based language models to resolve this weakness. The third weakness of *n*-gram models is due to the modeling of immediate history words where the information of long history words is neglected. In [6], long distance trigger pairs served as the basic elements for combination of multiple language sources. Bellegarda [2] exploited the latent semantic information for language models using large-span contexts.

Generally, the associated words appearing in the contexts may not contain only a pair of words but a sequence of words. It is unfeasible to incorporate the associations of three words together using the trigger pair language models [6][8]. Modeling of long distance word dependencies is restricted. In this paper, we concentrate on exploring the associations of more than two words so as to effectively resolve the insufficient long distance dependencies in *n*-gram models. The association patterns of multiple distant words are discovered during training. Due to the large span of multiple words, the proposed association pattern *n*-gram allows the semantic knowledge included in the language models. We amend the Apriori algorithm [1], which is popular in data mining field, to identify the association patterns. The mutual information of association patterns is measured and adequately contributed to the language modeling. To further overcome the problems of insufficient training data and domain knowledge, the association pattern *n*-gram is combined with the Witten-Bell smoothing for prediction of unseen word combinations and the cache mixture *n*-gram [5] for online unsupervised tracking of the evolutionary topics.

2. Data sparseness and domain mismatch

Regarding the issue of data sparseness, Witten-Bell smoothing [7] was shown to be effective in *n*-gram modeling [3]. Using this method, *n*-gram probability $p(w_i|w_{i-n+1}^{i-1})$ is smoothed by merging with (*n*-1)-gram $p(w_i|w_{i-n+2}^{i-1})$ through the recursive linear interpolation

$$p_{\mathrm{WB}}(w_i|w_{i-n+1}^{i-1}) = \eta_n p(w_i|w_{i-n+1}^{i-1}) + (1-\eta_n) p(w_i|w_{i-n+2}^{i-1}). \quad (1)$$

The factor $1-\eta_n$ stands for the frequency with which we should use (*n*-1)-gram to predict the next word. Let $N(w_{i-n+1}^{i-1}\cdot)$ denote the number of all possible words w_i adjacent to word sequence w_{i-n+1}^{i-1}. The number of occurrence of word combination of w_{i-n+1}^{i-1} and w_i is denoted by $c(w_{i-n+1}^{i-1}w_i)$. The interpolation factor has a form of

$$1-\eta_n = \frac{N(w_{i-n+1}^{i-1}\cdot)}{N(w_{i-n+1}^{i-1}\cdot) + \sum_{w_i} c(w_{i-n+1}^{i-1}w_i)}. \quad (2)$$

It has the interpretation of giving higher weight for (*n*-1)-gram under the case that more different words w_i are connected after word sequence w_{i-n+1}^{i-1}. When the possible words w_i adjacent to w_{i-n+1}^{i-1} are relatively few, it is preferable to rely on the contribution of *n*-gram. We only use (*n*-1)-gram when there are no occurrences for *n*-gram.

For the problem of domain mismatch, the degraded performance can be compensated via tracking the newest domain statistics at runtime. Here, we apply the cache mixture *n*-gram modeling where the model parameters are estimated in online unsupervised manner [5]. The ongoing updated *n*-gram is robust to changing topics in the documents. Applying the cache mixture *n*-gram, we incrementally adapt the model parameters sentence by sentence. The probability of word

sequence $W=\{W^1,\cdots,W^S\}$ composed of S sentences is characterized by using sentence-level mixture model containing m+1 components

$$p_{\mathrm{MX}}(W)=\prod_{s=1}^{S}p(W^s)=\prod_{s=1}^{S}\left[\sum_{k=1}^{m,G}\lambda_k^s\prod_{i=1}^{T_s}p_k(w_i\left|w_{i-n+1}^{i-1}\right.)\right], \quad (3)$$

where λ_k^s is the mixture weight estimated from preceding sentence W^{s-1} under the constraint $\sum_k\lambda_k^s=1$. And, $p_k(w_i\left|w_{i-n+1}^{i-1}\right.)$ denotes the k-th specific n-gram model trained from a different category of text documents. There are m mixture categories, which could be established either by supervised labeling or unsupervised clustering. The general n-gram model $p_G(w_i\left|w_{i-n+1}^{i-1}\right.)$ serves as the (m+1)-th mixture component, which covers non-topic content appearing in observed sentences. When the (s-1)-th sentence $W^{s-1}=\{w_1,\cdots,w_{T_{s-1}}\}$ is observed, we iteratively calculate the new mixture weight $\hat{\lambda}_k^s$ by maximizing the likelihood of W^{s-1} given the current estimate λ_k^s, i.e.

$$\hat{\lambda}_k^s=\frac{1}{T_{s-1}}\sum_{i=1}^{T_{s-1}}\frac{\lambda_k^s p_k(w_i\left|w_{i-n+1}^{i-1}\right.)}{\sum_{j=1,\cdots,m,G}\lambda_j^s p_j(w_i\left|w_{i-n+1}^{i-1}\right.)}. \quad (4)$$

To realize this algorithm, we prepare m+1 static n-gram models in training phase. When test data are gradually observed, the mixture weights λ_k^s are updated and applied to determine the likelihood $p(W^s)$ using the adapted language model. With the proper parameter λ_k^s, we are able to bring together the cache mixture n-gram $p_{\mathrm{MX}}(W)$, which continuously matches the newest topic knowledge in test article.

3. Modeling long distance word associations

Furthermore, the n-gram model is constrained by the insufficient modeling of associations longer than n words within or across sentences. Usually, the important semantic information is embedded in long distance words. It is crucial to detect long distance word associations and incorporate their information into language models. The trigger pair was chosen as the basic element for extracting information from the long distance document history [6][8]. In what follows, the trigger pair n-gram characterizing long distance language dependencies was described.

3.1. Trigger pair *n*-gram models

In natural language, if a trigger word w_i is significantly associated with a future word w_j, the trigger pair $w_i\rightarrow w_j$ is produced. The key issues of trigger pair n-gram aim at selecting and measuring trigger pairs. In trigger pair selection, we restrict the window size of two associated words so as to control the number of selected trigger pairs. Also, a simple way to measure the significance of the association is to measure the *average mutual information* (AMI) between words w_i and w_j [6][8]

$$\begin{aligned}&p(w_i,w_j)\log\frac{p(w_j|w_i)}{p(w_j)}+p(w_i,\overline{w}_j)\log\frac{p(\overline{w}_j|w_i)}{p(\overline{w}_j)}\\&+p(\overline{w}_i,w_j)\log\frac{p(w_j|\overline{w}_i)}{p(w_j)}+p(\overline{w}_i,\overline{w}_j)\log\frac{p(\overline{w}_j|\overline{w}_i)}{p(\overline{w}_j)}\end{aligned}, \quad (5)$$

where $p(w_i,\overline{w}_j)$ is the probability of occurring w_i but without w_j afterward in the window. AMI measures the information provided by w_i on w_j. A word pair is recognized as a trigger pair when its AMI is high. The set of trigger pairs $\Omega_{\mathrm{TR}}=\{w_i\rightarrow w_j\}$ can be selected from training corpus.

Assume there is a trigger pair $w_i\rightarrow w_j$ observed in word sequence $W=\{w_1,\cdots,w_i,\cdots,w_j,\cdots,w_T\}$, the conditional probability $p(w_j|w_i)$ should be considered in calculating the logarithmic probability using unigram models

$$\begin{aligned}\log p_{\mathrm{TR}}(W)&=\log\{p(w_1)\cdots p(w_i)\cdots p(w_j|w_i)\cdots p(w_T)\}\\&=\log\{p(w_1)\cdots p(w_i)\cdots p(w_j)\cdots p(w_T)\}+\log\frac{p(w_j|w_i)}{p(w_j)}\end{aligned}. \quad (6)$$

In (6), the second term represents the mutual information (MI)

$$\mathrm{MI}(w_i\rightarrow w_j)=\log\frac{p(w_j|w_i)}{p(w_j)}=\log\frac{p(w_i,w_j)}{p(w_i)p(w_j)}, \quad (7)$$

which reflects the degree of the preference for associations of w_i and w_j. If the events of occurring w_i and w_j are independent, then $\mathrm{MI}(w_i\rightarrow w_j)=0$. In practice, there exist several trigger pairs in word sequence W. We may express the trigger pair based observation probability as

$$\log p_{\mathrm{TR}}(W)=\sum_{i=1}^{T}\log p(w_i)+\sum_{i=1}^{T-1}\sum_{\substack{j>i,\,j-i\le ws\\ w_i\rightarrow w_j\in\Omega_{\mathrm{TR}}}}^{T}\mathrm{MI}(w_i\rightarrow w_j), \quad (8)$$

where ws denotes the predefined window size. We append the mutual information of all possible trigger pairs within window size $\{w_i\rightarrow w_j\in\Omega_{\mathrm{TR}},\,j-i\le ws\}$ to the estimation of $p_{\mathrm{TR}}(W)$. To improve the performance, we may combine the knowledge sources from the trigger pair model $p_{\mathrm{TR}}(W)$ and the static n-gram model $p(W)$. The trigger pair n-gram $\tilde{p}(W)$ is generated by [8]

$$\log\tilde{p}(W)=a_1\log p_{\mathrm{TR}}(W)+a_2\log p(W). \quad (9)$$

3.2. Association pattern *n*-gram models

Starting from the trigger pair n-gram, we present a new algorithm to construct the *association patterns* of more than two distant words and merge their mutual information into n-gram models. The selection of associated words is similar to the problem of discovering association rules between items in a large database of sales transactions, which has been extensively discussing in data mining community [1]. Data mining aims to discover all interesting rules from transaction databases. Such technology enables marketers to develop and implement customized marketing programs and strategies. In this paper, the *text database* is referred as the basket data for *mining association patterns of multiple words*. We are trying to efficiently identify all semantic patterns consisted of frequent associated words from training data. These patterns should exceed the predefined information-theoretic criterion.

The underlying concept of association pattern selection is to recursively *identify frequent word subsets* and *perform subset unification*. In the beginning, we scan the database and build the frequent one-word subset $L_1=\{w_i\}$. The frequency of each word is considered for selection. This subset has no association of words. To explore the frequent two-word subset $L_2=\{w_i\rightarrow w_j\}$, we unify different words in L_1 and generate the candidate two-word subset $C_2=\{w_i\bigcup w_j\}$. Frequent two-

word subset is selected from the candidate two-word subset, i.e. $L_2 \subseteq C_2$. The selection is based on the *average mutual information* $\mathrm{AMI}(w_i; w_j)$. We should scan all sentences and check four types of occurrences so as to calculate $p(w_i, w_j)$, $p(w_i, \overline{w}_j)$, $p(\overline{w}_i, w_j)$ and $p(\overline{w}_i, \overline{w}_j)$ for different word pairs in C_2. Those pairs $\{w_i \rightarrow w_j\}$ exceeding the minimum AMI form the subset L_2. We say the *association step* of L_2 is one because trigger word w_i only associates one word w_j. In a general selection procedure, the frequent a-word subset L_a is generated from the frequent (a-1)-word subset L_{a-1}. Let W_{a-1}^i denote an association pattern in $L_{a-1} = \{W_{a-1}^i\}$. We would like to use the sequence of a-1 words W_{a-1}^i as the *trigger sequence* to predict the occurrence of future word w_j. The frequent a-word subset $L_a = \{W_a^i\} = \{W_{a-1}^i \rightarrow w_j\}$ is established by fulfilling the following two passes.

1. **Join pass.** We scan the subset L_{a-1} and pick up the pattern W_{a-1}^j where the preceding a-2 words are identical to those of W_{a-1}^i. The last word w_j of pattern W_{a-1}^j is appended to W_{a-1}^i to generate the unification $W_{a-1}^i \bigcup w_j$. The candidate a-word subset $C_a = \{W_{a-1}^i \bigcup w_j\}$ is produced.
2. **Prune pass.** Two prune stages are performed. Fist, we delete the unification $W_{a-1}^i \bigcup w_j$ from C_a when some (a-1)-word subset of the a-word sequence $W_{a-1}^i \bigcup w_j$ is not in L_{a-1}. The candidate subset C_a can be refined to $\widetilde{C}_a$. To ensure the goodness of selection, we further prune the unification $W_{a-1}^i \bigcup w_j \in \widetilde{C}_a$ via evaluating the AMI between trigger sequence W_{a-1}^i and word w_j. The qualified association patterns $\{W_{a-1}^i \rightarrow w_j\} = \{W_{a-1}^i \bigcup w_j \in \widetilde{C}_a \,|\, \mathrm{AMI}(W_{a-1}^i; w_j) \geq \text{minimum AMI}\}$ are finally selected to form the frequent a-word subset L_a. These patterns involve a-1 association steps.

In this manner, the complete association pattern set Ω_{AS} covering different association steps are constructed by $\bigcup_{a=2}^{a_{\mathrm{up}}} L_a$. a_{up} is an upper bound for the number of words in the selected association patterns. Figure 1 shows the algorithm of finding association patterns in sentence level. Having the association pattern set Ω_{AS}, we merge the mutual information of all association patterns $W_{a-1}^i \rightarrow w_j$ into language modeling and yield the *association pattern model*

$$\log p_{\mathrm{AS}}(W) = \sum_{i=1}^{T} \log p(w_i) + \sum_{s=1}^{S} \sum_{W_{a-1}^{s,i} \rightarrow w_j^s \in \Omega_{\mathrm{AS}}} \mathrm{MI}(W_{a-1}^{s,i} \rightarrow w_j^s) \,. \quad (10)$$

The window size is set to be in sentence level. We search the occurrence of association patterns $W_{a-1}^{s,i} \rightarrow w_j^s$ within a sentence W^s. These selected words in association patterns are order dependent and semantically related. Similar to trigger pair n-gram, we estimate the *association pattern n-gram* by combining the association pattern model $p_{\mathrm{AS}}(W)$ and the static n-gram model $p(W)$ according to (9).

1) $L_1 = \{W_1^i\} = \{w_i \,|\, \text{frequent words}\}$;
2) $a = 2$;
3) **while** ($L_{a-1} = \{W_{a-1}^i\}$ is nonempty and $a \leq a_{\mathrm{up}}$) **do begin**
4) Apply *join pass* to generate preliminary candidates C_a.
5) Apply *prune pass* and refine candidates to $\widetilde{C}_a$.
6) **for all** sentences W^s in training corpus **do begin**
7) Find candidates $\{W_{a-1}^{s,i} \bigcup w_j^s \in \widetilde{C}_a\}$ contained in W^s.
8) Increment occurrence counts for these candidates.
9) **end**
10) Compute $\mathrm{AMI}(W_{a-1}^i; w_j)$ for all candidates in $\widetilde{C}_a$.
11) $L_a = \{W_a^i\} = \{W_{a-1}^i \rightarrow w_j\} = \{W_{a-1}^i \bigcup w_j \in \widetilde{C}_a \,|\, \mathrm{AMI}(W_{a-1}^i; w_j) \geq \text{minimum AMI}\}$;
12) $a = a + 1$;
13) **end**
14) $\Omega_{\mathrm{AS}} = \bigcup_{a=2}^{a_{\mathrm{up}}} L_a$;

Figure 1: Algorithm for mining association patterns.

Different from data mining algorithm [1], we use the information-theoretic AMI for selection of association patterns and the mutual information for measurement of word associations in language modeling. The mining algorithm is exploited to find the association patterns Ω_{AS} consisted of different numbers of associated words $\{L_2, L_3, \cdots, L_{a_{\mathrm{up}}}\}$. The proposed association pattern n-gram is a general framework where the mutual information between frequent a-1 word sequence W_{a-1}^i and associated word w_j is properly merged. In case of $a_{\mathrm{up}} = 2$, we build the relationship between trigger word w_i and associated word w_j. The association patterns become the frequent word pairs $\{w_i \rightarrow w_j\}$ which is also called the trigger pairs, i.e. $\Omega_{\mathrm{AS}} = \Omega_{\mathrm{TR}} = L_2$. Accordingly, the trigger pair n-gram is referred as a special realization of association-pattern n-gram in case of $a_{\mathrm{up}} = 2$.

4. Experiments

4.1 Experimental setup

To examine the performance of association pattern n-gram, we conduct a series of experiments and report the perplexities and speech recognition rates. Several databases were used. Dictionary was setup from the "Sinica corpus" with the size of five million Chinese words. We gathered 32941 frequent words to form the lexicon. Each word contained at most four Chinese characters. The articles in Sinica corpus were collected from different domains by the Institute of Information Science in Academia Sinica, Taiwan. This open source corpus was representative for Chinese language. We used this corpus as the training data to estimate n-gram models. Only bigram model was considered. In addition, we collected 3118 Chinese news documents covering eight categories: Technology, Society, Travel, World, Sports, Entertainment, Politics and Business. These classified documents were sampled from the news websites: CNA (http://www.cna.com.tw), ChinaTimes (http://news.chinatimes.com) and UDNnews (http://www.udnnews.com.tw), etc in Taiwan, during the period between April 10 and April 16 in 2001. We used Sinica corpus

and 2234 news documents (from April 10 to 14) for training and the remaining 884 news documents (April 15 and 16) for testing. Also, the experimental setup of speech recognition has been mentioned in [4]. Without loss of generality, the estimated language models were translated into syllable language models to perform syllable decoding of continuous speech. We reported the syllable recognition rates (%). The benchmark MAT-160 speech database was used to train speaker-independent HMM's. The test set (Test500) was recorded via telephones and consisted of 500 sentences from 30 outside speakers. It totally included 4754 syllables.

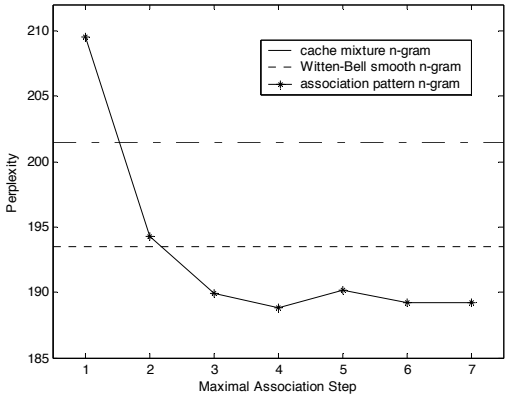

Figure 2: Comparison of different individual methods.

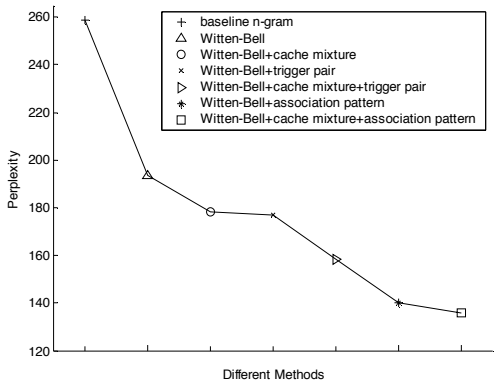

Figure 3: Comparison of different combined methods.

4.2 Experimental results

When the association pattern *n*-gram is implemented, we specify the maximal association step and apply the association pattern mining algorithm to determine all patterns covering different association steps. The association patterns are recursively extracted from single association step to maximal association step. In Figure 2, we compare the perplexities of cache mixture *n*-gram, Witten-Bell smooth *n*-gram and association pattern *n*-gram under different maximal association steps. Trigger pair *n*-gram is a special case of association pattern *n*-gram with maximal association step being two. This figure indicates that the association pattern *n*-gram is better than cache mixture *n*-gram and Witten-Bell *n*-gram when involving more than two association steps. In case of four association steps, the lowest perplexity 188.8 is attained. In the subsequent experiments, we only report the association pattern *n*-gram with maximal association step being four. Basically, these three methods are developed to resolve different problems. These methods can be combined to improve language modeling performance. In Figure 3, we investigate the perplexities of different combined approaches. Among all combinations, the lowest perplexity 135.8 is achieved when Witten-Bell smoothing, cache mixture *n*-gram and association pattern *n*-gram are simultaneously performed. Also, the proposed language models are applied for continuous Mandarin speech recognition. Language models are merged to speech recognition system in syllable level. The experiments show that syllable recognition rates are increased from 51.3% without language model to 55.8% with Witten-Bell smooth *n*-gram. If trigger pair *n*-gram and association pattern *n*-gram are fulfilled, syllable recognition rates are improved to 56.7% and 57%, respectively. These two results are comparable because the lengths of test utterances are not too long.

5. Conclusion

This paper have surveyed three essential problems in statistical *n*-gram and presented the hybrid approaches to achieve sophisticated language modeling. The techniques of Witten-Bell smoothing, cache mixture *n*-gram and trigger pair *n*-gram were introduced to cope with the problems of data sparseness, domain mismatch and long distance dependency, respectively. To relax the constraints of trigger pair *n*-gram characterizing the information of two distant words, this paper explored a novel association pattern *n*-gram where the word associations of the frequent word sets consisted of more than two distant words were merged in *n*-gram. The frequent word sets, also called the association patterns, were determined through the association pattern mining algorithm. We consistently used information-theoretic criterion and measure. The averaged mutual information criterion was applied to judge whether the selected word sets are frequent or not. Also, the mutual information of association pattern was measured for association pattern *n*-gram. From the experimental results, we find that the association pattern *n*-gram is better than Witten-Bell smoothing, cache mixture *n*-gram and trigger pair *n*-gram. The performance of combined methods can be further improved.

6. References

[1] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules", *Proceedings of the International Conference on Very Large Data Bases (VLDB)*, pp. 487-499, 1994.

[2] J. R. Bellegarda, "Exploiting latent semantic information in statistical language modeling", *Proceedings of the IEEE*, vol. 88, no. 8, pp. 1279-1296, 2000.

[3] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling", *Computer Speech and Language*, vol. 13, pp. 359-394, 1999.

[4] J.-T. Chien, C.-H. Huang and S.-J. Chen, "Compact decision trees with cluster validity for speech recognition", *ICASSP*, pp. 873-876, 2002.

[5] R. M. Iyer and M. Ostendorf, "Modeling long distance dependence in language: topic mixtures versus dynamic cache models", *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 1, pp. 30-39, 1999.

[6] R. Lau, R. Rosenfeld and S. Roukos, "Trigger-based language models: a maximum entropy approach", *ICASSP*, pp. 45-48, 1993.

[7] I. H. Witten and T. C. Bell, "The zero-frequency problem: estimating the probabilities of novel events in adaptive text compression", *IEEE Transactions on Information Theory*, vol. 37, pp. 1085-1094, 1991.

[8] G. D. Zhou and K. T. Lua, "Interpolation of *n*-gram and mutual-information based trigger pair language models for Mandarin speech recognition", *Computer Speech and Language*, vol. 13, pp. 125-141, 1999.