interspeech 2008 incorporating SST 08

# Improving Japanese Language Models Using POS Information

*Langzhou Chen* [1], *Hisayoshi Nagae* [2] *and Matt Stuttle* [1]

[1] Toshiba Research Europe Limited, 208, Science Park, Cambridge, UK
[2] Toshiba Corporate Research & Development Center, Kawasaki, 212-8582, Japan
lchen@crl.toshiba.co.uk, hisayoshi.nagae@toshiba.co.jp

## Abstract

In this paper, part-of-speech (POS) information is used to improve the performance of a Japanese language model (LM). The POS bigram is used to tackle the sparseness problem of the training data. Additionally, due to the characteristics of the Japanese language, part of the Japanese syntax information can be integrated into the POS bigram, through POS combination rules. Based on the Japanese syntax grammar, the POS combination rules determine if a POS pair is prohibited in Japanese language. The Japanese POS bigram table not only includes the POS pairs that occurred in the training corpus, but also includes all the prohibited POS pairs. The confusion in the search space can be reduced by explicitly modeling the prohibited POS pairs. In this work, a series of experiments have been carried out to investigate the impact of the POS bigram with prohibited POS pairs on the recognition search space. The framework of fast generation of the language model look-ahead (LMLA) probabilities based on POS bigram information is also presented in this paper. The experimental results showed that compared to the traditional word n-gram model, the LM with POS bigram information achieves significant improvement in both word accuracy and the speed of Japanese LVCSR system.

**Index Terms**:language model, Speech Recognition, decoding

10.21437/Interspeech.2008-250

## 1. INTRODUCTION

Data sparsity is a recurring problem in generating robust language models (LMs) for speech recognition. Class based LMs[1] have been widely used to solve the data sparseness problem. The words in the vocabulary are divided into classes, and a class based n-gram is estimated to approximate the word based n-gram probabilities. The class-based LM can be combined with the word based n-gram by linear interpolation [2]. Another popular way to smooth the word n-gram model is the class based backoff model. If the word n-gram can not be found in LM data, the class based n-gram is used as backoff information. In order to improve the accuracy of the class-based model, some more complicated backoff strategies have been proposed. For example, the backoff hierarchical class n-gram model was constructed through clustering the vocabulary hierarchically into a word tree and the probabilities of unseen events were always estimated by the most specific class of the tree [3]. In order to introduce more syntax information into a class based LM, POS information has been used in class based n-gram modeling. In [4], the vocabulary words are divided into 25 POS tags and the word clustering carried out based on different POS tags.

In the work of this paper, Japanese POS tags are adopted in a class based n-gram model. The POS tags used in this work not only include the grammatical definition of the word, but also the context information between words. Traditional class based n-gram models only contain the statistical information from training corpus. However, because of the characteristics of the Japanese language, the syntax information which is not included in the training corpus can be incorporated into the POS based n-gram model.

Based on linguistic knowledge of the Japanese language, a list of POS combination rules was generated. The POS combination rules contain the Japanese syntax information which is independent of the training corpus. The POS combination rules are integrated into POS n-gram data by an indicator function. Therefore, the POS n-gram table contains not only the statistical information from training corpus but also the prohibited POS pairs that can never occur in Japanese. In the traditional class based n-gram model, the prohibited POS pairs are treated as unseen events and their probabilities are estimated using a lower order LM. Consequently, hypotheses with prohibited POS pairs can still survive in the search space. On the other hand, when the prohibited POS pairs are explicitly modeled, hypotheses with prohibited POS pairs will be pruned from the search space immediately. This means that the POS combination rules use syntax information to achieve efficient pruning of the search space. This is different from some well known methods which use the syntax as long distance information to compensate the n-gram model [5].

In this paper, a series of experiments are reported to investigate how the POS n-gram with prohibited POS pairs influences the recognition search space. Different pruning levels are used to test the impact of the POS n-gram on both word accuracy and decoding speed.

In this paper, we also present the method to generate the LM look-ahead (LMLA) efficiently based on the POS backoff n-gram model. An efficient method to generate LMLA probabilities has been presented based on word n-gram LMs [6]. It is extended here to the POS backoff n-gram LM. The rest of this paper is organised as follows: in section 2, the Japanese POS backoff LM is presented; in section 3, the efficient method for calculating the LMLA probabilities based on POS backoff LM is introduced. Experimental results are presented in section 4. Finally, the conclusions are given.

## 2. POS BASED N-GRAM FOR JAPANESE SLM

### 2.1. POS backoff LM

Class based LMs are a widely used technique to overcome the LM training data sparseness problem. The words in the vocabulary are partitioned into classes. If the word sequence did not occur in the training data, then the inter-word probabilities are estimated by inter-class probabilities. When the training data is large, the class based n-gram model should work together with word n-gram to get better performance. A

popular way to integrate word n-gram and class based n-grams is the Backoff model.

$$P(w_i \mid w_{i-n+1,} \cdots w_{i-1}) = \begin{cases} f(w_i \mid w_{i-n+1,} \cdots w_{i-1}) & if(N(w_{i-n+1} \cdots w_i) > 0) \\ P(w_i \mid C_i) * P(C_i \mid C_{i-n+1}, \cdots, C_{i-1}) * & \\ Bo(w_{i-n+1,} \cdots w_{i-1}) & otherwise \end{cases} \quad (1)$$

Where $N(*)$ is the frequency event '*' occurred in training data, and $C_i$ is the class tag of word $w_i$. In this work, we also use Eqn 1 to integrate the class n-gram information.

Automatic word clustering has been used successfully in class based LMs, however, it is influenced by the biased information of the training corpus. On the other hand, the POS information is independent of the training data and contains more syntax information of the natural language. Therefore, in this work, the words in the vocabulary are clustered based on their POS tags. If only considering the grammatical definitions of the words, the number of POS tags is quite small, for example, in [4], the words are tagged into 25 POS classes. The POS tags used in this work not only includes the word definition information, but also the connection information between words. Therefore, the number of the POS tags is much bigger.

### 2.2. POS combination rules

The Japanese language has some special characteristics. Combinations of POS pairs are restricted in Japanese grammar. For example, each Japanese function word can only be connected with a specific set of POS tags, while the remaining combinations are grammatical errors in Japanese. Therefore POS combination rules can be created to describe the syntax information of Japanese language. The POS combination rules are represented as a binary indicator function, i.e. $I(C_i, C_j)=1$, if connection of "$C_i$ , $C_j$" is true in the POS combination rules, otherwise, $I(C_i , C_j )=0$. The probabilities in the POS n-gram are weighted by the indicator function, i.e.

$$P(C_i \mid C_{i-n+1} \ldots C_{i-1}) = f(C_i \mid C_{i-n+1} \ldots C_{i-1}) * I(C_{i-1,} C_i) \quad (2)$$

Where $f(C_i \mid C_{i-n+1} \ldots C_{i-1})$ is the estimated value of the POS n-gram model.

In the traditional class based n-gram model, the prohibited POS pairs are treated as unseen events and their probabilities are estimated using a lower order LM. Consequently, the hypotheses with prohibited POS pairs can still survive in the search space. When the POS combination rules are adopted, the hypotheses with prohibited POS pairs will be pruned from the search space immediately. Therefore introducing the POS combination rules into a class based LM can reduce the confusion of the recognition search space.

## 3. FAST LMLA BASED ON POS BACKOFF LM

The calculation of LMLA influences the speed of the decoding process directly. It is important to minimise the calculations for LMLA. We have previously presented a method to accelerate the LMLA process based on a word based n-gram [6]. In this paper, this method is extended to the POS backoff LM.

### 3.1. Generating LMLA probabilities using low order LML trees

In the new LMLA probability generation method, the higher order LMLA probabilities are calculated from the lower order LMLA trees. The method takes advantage of the sparseness of the n-gram LM to avoid unnecessary computation. In a backoff based LM, given the word context information, only a small part of n-gram probabilities are estimated explicitly, while the rest are backoff estimates. This fact led to the development of a LMLA probabilities generation algorithm that achieves its goal by updating a (n-1)-gram LMLA tree. Only nodes that are related to explicitly estimated n-gram values are updated, the rest of the nodes are the backoff of the corresponding nodes in the (n-1)-gram LMLA tree. Since the number of nodes which relate to the explicit n-gram probabilities are much smaller than the total number of nodes in the LMLA tree, the new method significantly reduces the LMLA calculation.

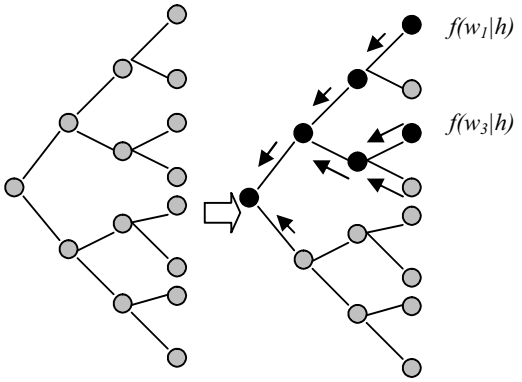


Figure 1: *Generating the LMLA probability using lower order LMLA tree.*

Figure 1 shows the calculation of LMLA probabilities based on the lower order LMLA tree. The LMLA tree in Figure 1 contains 8 leaves, i.e. 8 individual words. Given the LM context $h$, supposing that only 2 words: $w_1$ and $w_3$, have explicit LM probabilities, the new method only needs to calculate the LMLA probabilities in the nodes from which words $w_1$ and $w_3$ can be reached, i.e. the black nodes in Figure 1, while the rest of the LMLA probabilities, i.e. the LMLA probabilities in the grey nodes, can be copied from the backoff LMLA tree directly.

### 3.2. Generating LMLA probabilities based on POS backoff LM

The LMLA calculation method presented above can be extended to the POS backoff LM. For the bigram, the POS backoff model can be expressed as

$$P(w_1 \mid w_2) = \begin{cases} f(w_1 \mid w_2) & if(C(w_2, w_1) > 0) \\ P(w_1 \mid C_1) * P(C_1 \mid C_2) * Bo(w_2) & otherwise \end{cases} \quad (3)$$

Given a POS tag $C_i$ , and a node $n$, the LMLA probability can be expressed as

$$\pi(n \mid C_i) = \max_{w \in W(n)} P(C(w) \mid C_i) * P(w \mid C(w)) \quad (4)$$

where $W(n)$ represents the set of the words that can be reached from node $n$ and $C(w)$ is the POS tag of word w. Figure 2 shows the framework of efficient LMLA probability generation using a POS backoff LM. Based on Eqn 4, for every POS history, using $C_i$ as index, we can find the POS bigram LMLA tree for $C_i$ from the POS bigram LMLA tree buffer. Based on the LMLA tree for $C_i$ , we only need to update the probabilities for the nodes which are related to the explicit word bigrams in the LM data, to generate the LMLA tree for word $w_i$.

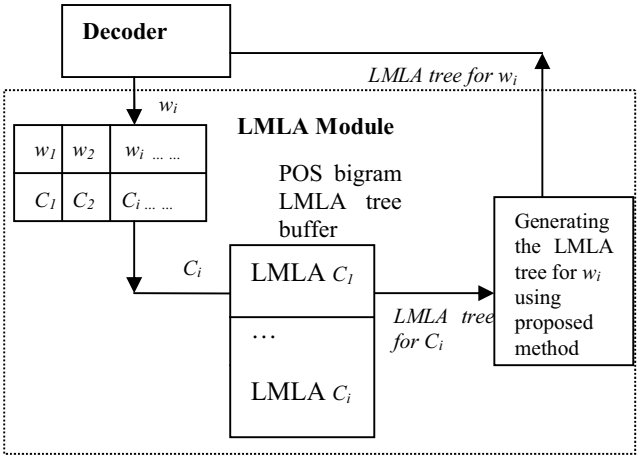


Figure 2: the framework of efficient LMLA probability generation using POS backoff LM.

### 3.3. Reducing LMLA calculation cost by LM pruning

The new method of fast calculation of the LMLA probabilities makes use of the data sparseness of the language model data. Given a history word $h$, defining $Node(w)$ as the set of nodes from which the word $h$ can be reached, all the nodes in the LMLA tree whose LMLA probability need to be updated can be expressed as

$$Node\_update = \bigcup_{\substack{P(w|h)exist \\ in\ LM\ data}} Node(w) \quad (5)$$

Therefore, the calculation cost of the LMLA depends on how many word bigrams the LM data explicitly contains. The less the number of explicitly estimated word bigrams are stored in the model, the less the calculation cost of LMLA is. The entropy pruning method [7] is adopted to reduce the number of explicit word bigrams in the model. Only word pairs (h,w) that satisfy the following criterion are stored in the LM data,

$$\sum_{w} P(w,h)[\log P(w|h) - \log P'(w|h)] > \theta$$
$$P'(w|h) = P(C(w)|C(h)) * P(w|C(w)) * Bo(h) \quad (6)$$

where $\theta$ is an empirical threshold and $C(w)$ is the POS tag of word w. The same pruning can be carried out as in the word n-gram model, i.e. the word unigram backoff estimate is used to replace $P'(w|h)$ in Eqn6. However, the experimental results showed that the POS bigram backoff model is much closer to the word bigram model compared to the word unigram backoff model. Therefore, using a POS bigram backoff model, the number of explicitly estimated word bigrams is reduced significantly, as is the calculation of the bigram LMLA.

## 4. EXPERIMENTAL RESULTS

The experiments of POS backoff LM reported in this paper are based on a Japanese 40k vocabulary dictation system. The training speech contains about 130 hours speech data. The size of vocabulary is about 42k words. The acoustic model contains 3000 tied HMM states with 20 Gaussian mixture components per state. The speech feature vector is 33 dimensions, including 10 MFCC, 1 LOG energy, and their first-order and second-order time derivatives. The LM training corpus contains 2 years Yomiuri Shimbun Japanese newspaper database, about 10M sentences. Two LMs were trained in the experiments: word based trigram model; word trigram with POS bigram backoff model. The test set contained 1000 Japanese sentences relating to travel.

Table 1 shows the comparison results between the word based n-gram and the POS backoff n-gram. The results for word error rate (WER), character error rate (CER) and the real time factor (RTF) of the recognizer are presented.

| **Word trigram** | | | **POS backoff model** | | |
|---|---|---|---|---|---|
| RTF | WER | CER | RTF | WER | CER |
| 0.80 | 25.5% | 17.3% | 0.88 | 19.9% | 13.5% |
| 0.73 | 26.1% | 17.8% | 0.79 | 20.3% | 13.8% |
| 0.68 | 27.0% | 18.4% | 0.72 | 21.1% | 14.5% |
| 0.59 | 28.2% | 19.5% | 0.63 | 22.0% | 15.4% |
| 0.49 | 30.9% | 21.4% | 0.53 | 24.3% | 17.5% |

Table 1. *Comparison of results between word based n-gram and POS backoff n-gram*

In Table 1, every row represents the results based on the same beam width. The trigram model with POS bigram backoff can be seen to work consistently better than the traditional word based trigram over each beam width. To achieve a similar recognition accuracy, the system with POS backoff LM is much faster than the system with traditional n-gram.

In Table 2 some details of the search spaces are presented. For this comparison, the two systems were tuned separately to achieve a similar recognition rate.

| | **Word n-gram** | **POS backoff model** |
|---|---|---|
| # of active LMLA trees | 351 | 139 |
| Avg. # of active states | 1375 per frame | 501 per frame |
| Avg. # of active hyp. | 5.0k per frame | 1.9k per frame |
| CER | 17.0% | 16.1% |
| WER | 25.0% | 23.0% |
| RTF | 0.89 | 0.58 |

Table 2. *Comparison of search space between word based n-gram and POS backoff n-gram*

Table 2 shows 3 parameters of the search space, the first line is the number of active LMLA trees in the search space, the second line is the number of average active HMM states

per frame and the third line is the number of average active hypotheses per frame. Based on similar recognition accuracy, the search space of the POS backoff LM is much smaller than the word n-gram search space. This yields a reduction in the real time factor of the system. The system based on the POS backoff model is 35% faster than word n-gram system.

As discussed in section 2, the POS combination rules were introduced to reduce confusions in the search space. Therefore the Japanese POS backoff model takes the advantages of both the good smoothing from the class-based model and the syntax information from the POS combination rules. It is interesting to investigate these 2 factors separately, to see how much contribution is provided by each factor. Two POS backoff models were generated, one with the POS combination rules, the other without the POS combination rules. The experimental results are shown in Table 3.

| | **Without POS combination rules** | **With POS combination rules** |
|---|---|---|
| # of active LMLA trees | 175 | 139 |
| Avg. # of active states | 503 per frame | 501 per frame |
| Avg. # of active hyp. | 2.5k per frame | 1.9k per frame |
| CER | 15.9% | 16.0% |
| WER | 22.9% | 23.0% |
| RTF | 0.68 | 0.58 |

Table 3. *Comparison results between the model with POS combination rules and without the POS combination rules*

Table 3 shows that the two systems achieve similar recognition rates. However the system with POS combination rules is faster than the system without POS combination rules by 14.7%. The search space of the model with POS combination rules can also be seen to be smaller. This result shows the effectiveness of the POS combination rules in pruning the decoding space.

| $\theta$ | **Word n-gram** | | **POS backoff model** | |
|---|---|---|---|---|
| | **# bigrams** | **# of updated nodes** | **# bigrams** | **# of updated nodes** |
| 0 | 1.6M | 7309 | 1.6M | 7309 |
| $1.0\times10^{-10}$ | 1.5M | 7225 | 1.6M | 7240 |
| $1.0\times10^{-9}$ | 1.3M | 7111 | 1.5M | 7207 |
| $1.0\times10^{-8}$ | 0.9M | 6703 | 1.2M | 7020 |
| $5.0\times10^{-8}$ | 0.5M | 4724 | 0.9M | 6634 |

Table 4. *Impacts of LM pruning on the calculation quantities of LMLA*

As discussed in section 3, the new method of generating the LMLA probabilities depends on the number of explicit n-grams which are stored in the LM data. Table 4 shows the impact of LM pruning on the calculation quantity of LMLA. Entropy-based pruning is adopted to prune the LM data, as this method has been proven to be very effective to compress LMs. In this experiment, we don't focus on the size of LM data, but on the calculation quantity of LMLA.

Table 4 shows the average number of nodes which needed to be updated in the LMLA tree at different pruning thresholds, $\theta$,. When the size of the bigram data reduced, the number of the nodes that needed to be updated in LMLA tree also decreased. However, the POS bigram provided more accurate backoff estimation than the word unigram. Therefore, based on the same pruning threshold, the POS backoff model contains slightly less explicit bigrams and the calculation cost for LMLA is also less than word n-gram.

# 5. CONCLUSIONS

In this paper, a language model was proposed using POS tags for classes in a class based n-gram. The structure of the Japanese language was exploited to add an additional constraint to the LM in terms of a set of POS combination rules. The Japanese POS backoff language model takes advantage of both the better smoothing from the class-based n-gram model and the syntax information from the POS combination rules. Experimental results showed that the POS backoff LM works better than a word based n-gram both in terms of word accuracy and speed. The POS combination rules in particular reduce the search space and confusions within. Additionally, a framework to quickly generate the LMLA probabilities based on POS backoff LM was presented.

# 6. REFERENCES


[1] F. Jelinek, "Self-organized language modeling for speech recognition," Reading in speech recognition, Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1990.

[2] Gilles Adda, Michéle Jardino and Jean-Luc Gauvain, "Language modeling for broadcast news transcription", In Proceedings of EUROSPEECH'99, pp.1759-1762.

[3] Imed Zitouni, "Backoff hierarchical class n-gram language models: effectiveness to model unseen events in speech recognition," Computer Speech and Language, 21, 2007, pp.88-104

[4]Wen Wang and Dimitra Vergyri, "The use of word n-grams and parts of speech for hierarchical cluster language modeling," In Proceedings of ICASSP 2006, Vol 1, pp1057-1060

[5]C.Chelba and F.Jelinek, "Recognition Performance of a structured language model," in Proceedings of EUROSPEECH, VOL.4, 1999, pp. 1567-1570

[6] L. Chen and K.K. Chin, "Efficient language model look-ahead probabilities generation using lower order LM look-ahead information," In Proceedings of ICASSP 2008.

[7] A. Stolcke, "Entropy-based Pruning of Backoff Language Models," In Proceedings of DARPA broadcast news transcription and understanding workshop, Lansdowne, pp.270-274, 1998