Vol.2009-SLP-79 No.8
2009/12/21

Topic Dependent Language Model based on On-Line Voting

Welly Naptali, Masatoshi Tsuchiya, and Seiichi Nakagawa[†1,†2,†1]

In this paper, we propose an alternative approach to a topic dependent language model (LM), where the topic is decided by voting in an unsupervised manner. Latent Semantic Analysis (LSA) is employed to reveal hidden (latent) relations among nouns in the context word sequence. To decide the topic of an event, a fixed size word history sequence (window) is observed, and voting is then carried out based on noun class occurrences weighted by a confidence measure. Experiments on the Wall Street Journal corpus and Mainichi Shimbun (Japanese newspaper) corpus show that our proposed method gives better perplexity than the comparative baselines, including a word-based/class-based n-gram LM, their interpolated LM, a cache-based LM, and the Latent Dirichlet Allocation (LDA)-based topic dependent LM.

1. Introduction

A statistical language model (LM) plays an important role in automatic speech recognition (ASR) systems. It has been used to reduce the acoustic search space and resolve acoustic ambiguity. Statistical n-gram LMs are very good at modelling short-range dependencies, but not at modelling long-range dependencies. Several attempts have been made to capture long-range dependencies. The cache-based model[10] increases the probability of words observed in the history. The trigger model[12] is a generalization of the cache-based model, but its training is computationally very expensive. A topic mixture model[8] tries to capture topic-related constraints within and between sentences by combining a number of word-based n-gram LMs trained on topic-specific documents. Bellegarda[1] combined the global constraint given by Latent Semantic Analysis (LSA) with the local constraint of the n-gram language model. This success was followed by the Probabilistic LSA (PLSA)[7] and Latent Dirichlet Allocation (LDA)[3] models. More recently, a topic dependent LM using LDA to detect topics has been proposed[11]. However, this method needs pre-analysis of the test dataset to determine topic mixtures, thus it is impossible to apply this method against a real-time ASR system.

In this paper, we propose an alternative topic dependent LM in which the topic of an event is decided by voting. First, LSA is applied to map nouns into a semantic vector space, and then a *vector quantization* (VQ) is conducted to define the topics. The distance between a noun vector and the topic centroid is defined as the confidence measure. A fixed size word history sequence (window) is observed to decide the topic of an event. Unlike other topic dependent LMs, the topic is integrated as a part of the word sequence in the n-gram model.

The rest of the paper is organized as follows: Section 2 describes the process of defining topics and the confidence measure. Section 3 explains how the topic of an event is decided. Section 4 defines the proposed LM, the topic dependent class (TDC) in detail. Section 5 describes the experiments and gives the performance results for the proposed model as well as several baselines for comparison. The paper ends with the conclusions and possible future works.

2. Topics and Confidence Measure

Nouns in a sentence play an important role in the whole discourse and are the core of the underlying LM. The associations between nouns are supporting factors in defining topics. We use LSA to reveal these hidden relations to form topics. LSA extracts semantic relations from a corpus, and maps them into a semantic vector space. The discrete indexed words are projected into the LSA space by applying *singular value decomposition* (SVD) to a matrix representing the corpus. A noun-document matrix is used to represent the training corpus in which the rows correspond to nouns and the columns to documents. We apply a *term frequency-inverse document frequency* (tfidf) weight to each noun in each document. Once the vector representation for each noun has been created in the LSA space, VQ[5] is applied to cluster these words into topics. VQ is performed using the cosine similarity between nouns until the desired number of clusters (topics) is reached. A confidence measure γ is defined as the distance between a

†1 Department of Information and Computer Sciences, Toyohashi University of Technology
†2 Information and Media Center, Toyohashi University of Technology

word vector and its class centroid. So in this case, $\gamma(w_i)$ can be calculated using the same cosine similarity between noun w_i and its topic class C_i,

$$\gamma(w_i) = cos(\mathbf{w_i}, \mathbf{C_i}) = \frac{\mathbf{w}_i.\mathbf{C}_i}{|\mathbf{w}_i||\mathbf{C}_i|}, \quad (1)$$

where $\mathbf{w}_i$ is the word vector mapped into the LSA space of word w_i, and $\mathbf{C}_i$ is the centroid vector of topic class C_i. This score ($0 \leq \gamma \leq 1$) indicates how confident a noun w_i is to be in the topic class C_i. The larger the γ score, the more typical a word w_i is in the class C_i.

3. Topic Voting on Noun History

Given the history of a word sequence $w_1, w_2, ..., w_{i-1}$, the topic of this particular event in an n-gram model is denoted as $Z_{i-n}(m)$, where m is the window size. Topic $Z_{i-n}(m)$ is obtained by observing m words in the outer contexts of the near n-gram $w_{i-n-m+1}, ..., w_{i-n}$ by voting on the weighted frequency of the noun classes. Mathematically this can be written as

$$Z_{i-n}(m) = \arg\max_{C} \sum_{j=i-n-m+1}^{i-n} \gamma(w_j)\delta(C_j, C), \quad (2)$$

where $\delta(C_i, C) = \begin{cases} 1 & \text{if } C = C_i, \\ 0 & \text{otherwise.} \end{cases}$

Note that γ and δ are defined only for nouns, otherwise 0 is assigned. If there are no nouns inside window $Z_{i-n}(m)$, as happens at the beginning of a document, a dummy topic class $C_\emptyset$ is defined.

4. Topic Dependent Class LM

Humans may not be able to understand a conversation in which the topic is unknown. In the same sense, it is easier for an LM to predict the current word w_i if the topic is known since this reduces the search space of possible candidates for the current word. A topic dependent class (TDC) is proposed to provide such information to the LM. A TDC with window size m is an LM in which the probability of a word sequence $W = w_1, w_2, ..., w_N$ is calculated according to:

$$P_{TDC}(W) = \prod_{i=1}^{N} P(w_i | Z_{i-n}(m), w_{i-1}^{i-n+1}), \quad (3)$$

where $w_{i-1}^{i-n+1} = w_{i-n+1}, ..., w_{i-1}$ and $Z_{i-n}(m)$ is the topic class described in the previous section.

Sometimes it could be dangerous when deciding a word sequence belongs to only one topic (hard-voting). A word sequence may usually belong to multiple topics. Based on this idea, we also perform a soft-decision on voting (soft-voting) in the test phase. Instead of choosing the best topic in Eq. (2), we choose K-best topics and then interpolate them linearly. So Eq. (3) becomes

$$P_{TDC}(W) = \prod_{i=1}^{N} \sum_{j=1}^{K} \alpha_j P(w_i | Z_{i-n,j}(m), w_{i-1}^{i-n+1}), \quad (4)$$

where mixture weight α is calculated in according to $\alpha_j = \frac{\beta_j}{\sum_{k=1}^{K} \beta_k}$, where β is the score that was obtained during voting (see Eq. (2)).

As we can see, the topic is integrated within the word based n-gram LM. The equation is similar to the factored LM[2)] in which the last context is considered as the topic $Z_{i-n}(m)$. However, the topic $Z_{i-n}(m)$ in this case is not decided based on a specific word w_{i-n}, but on the collection of word history instead. A statistical word-based 3-gram LM is used to capture the local constraint using linear interpolation.

4.1 Backoff for Unseen Events

In an n-gram LM, when the model encounters unseen events, it is usually backed-off by the shorter $(n-1)$-gram. In our model, we follow a similar approach in handling unseen events. We use the Katz backoff with an absolute discounting method. If the sequence $Z_{i-n}(m), w_i^{i-n+1}$ is seen in the training dataset, or $f(Z_{i-n}(m), w_i^{i-n+1}) > 0$, then

$$P_{TDC}(w_i | w_{i-1}^{i-n+1}) = P'(w_i | Z_{i-n}(m), w_{i-1}^{i-n+1}), \quad (5)$$

otherwise

$$P_{TDC}(w_i | w_{i-1}^{i-n+1}) = \alpha(Z_{i-n}(m), w_{i-1}^{i-n+1}) P'(w_i | Z_{i-n+1}(m), w_{i-1}^{i-n+2}), \quad (6)$$

where P' is the discounted probability and α is the backoff weight. Note that the backoff method is not performed by eliminating word w_{n-1} in our model, but by sliding the window $Z_{i-n}(m)$ to $Z_{i-n+1}(m)$. The backoff weight α is estimated as $\alpha(Z_{i-n}(m), w_{i-1}^{i-n+1}) = \frac{1-\sum_{w_i:f(Z_{i-n}(m),w_{i-1}^{i-n+1})>0} P'(w_i|Z_{i-n}(m),w_{i-1}^{i-n+1})}{1-\sum_{w_i:f(Z_{i-n}(m),w_{i-1}^{i-n+1})>0} P'(w_i|Z_{i-n+1}(m),w_{i-1}^{i-n+2})}$. The history $\{Z_{i-n+1}(m), w_{i-1}^{i-n+2}\}$ on $P'(w_i|Z_{i-n+1}(m), w_{i-1}^{i-n+2})$ is a shorter history of $\{Z_{i-n}(m), w_{i-1}^{i-n+1}\}$. This is done by sliding the window, so that window $Z_{i-n}(m)$ releases the last word $w_{i-n-m+1}$, and adding word w_{i-n+1} into the window to become $Z_{i-n+1}(m)$. Recording these kinds of events is computationally expensive and therefore, as the window size is quite large, a topic switch happens rarely. The assumption is thus made that such a word exchange does not affect the topic too much[⋆1], or $Z_{i-n+1}(m) \approx Z_{i-n}(m)$.

5. Experiments and Results

Here we compare our proposed method with several baseline methods, namely a word-based/class-based LM, a cache-based LM, and an LDA-based topic dependent LM. The comparison is based on perplexity $PP = 2^{-\frac{1}{N}\log_2 P_{LM}(W)}$. To decompose the matrix representation, we use the SVDLIBC toolkit[⋆2]. All nouns in this experiment are mapped to a 200 dimensional LSA space. Then VQ clustering is carried out in this LSA space using the Gmeans toolkit[6] for a given number of topics. In this study, we only conducted TDC 3-gram model. No pruning is applied in the TDC LM, nor in the baseline methods. For each model, an interpolated model is defined as linear interpolation given by $P_{LM} \approx \lambda P_{LM1} + (1-\lambda)P_{LM2}$, where λ is the interpolation weight constant optimized on $(0 < \lambda < 1)$ with stepsize 0.1. In this paper, LM2 is a word-based 3-gram, while LM1 could be class-based, cache-based, LDA-based, or the proposed method. For a cache-based LM, since it is trained based on only a limited word history, the interpolation weight λ is usually very small. We used λ optimized on $(0 < \lambda < 0.2)$ with stepsize 0.01.

⋆1 This assumption makes the total probability of some word sequences, that has backed-off probability does not always equal to 1. A normalization procedure can be conducted, but it will be time consuming as it has to calculate the probability of all words in the vocabulary for a given word sequence. We confirmed that the perplexity resulting from the normalized probability and the unnormalized probability is almost similar. As the window size increasing, the differences is unnoticeable.

⋆2 http://tedlab.mit.edu/dr/svdlibc

Table 1 TDC and statistical n-gram LM perplexity for WSJ corpus.

No	Model		PP
1	Word-based	3-gram	111.6
2		4-gram	101.4
3	Class-based	3-gram	132.4
4	(class=2000)	4-gram	121.5
5	Class-based 3-gram + Word-based 3-gram ($\lambda = 0.3$)		106.4
6	Class-based 4-gram + Word-based 3-gram ($\lambda = 0.4$)		**98.9**

5.1 WSJ Corpus

The data, taken from the Wall Street Journal (WSJ) corpus between the years 1987 and 1989, were divided into training and test datasets. The training dataset contains 36,754,891 words in 85,445 documents, while the test dataset contains 336,096 words in 809 documents. ARPA's official "20o.nvp" (20k most common WSJ words with non-verbalized punctuation) is used as vocabulary, and it gives OOV rate 2.47% and 2.57% for the training and test datasets, respectively. By adding a beginning sentence symbol <s>, an end sentence symbol </s>, and an unknown symbol <UNK> to map all out-of-vocabulary (OOV) words, the total vocabulary size is 19,982 words. To filter nouns in the vocabulary, we used the TreeTagger toolkit[13].

5.1.1 Statistical n-gram LM

The word-based n-gram LM is the most common LM currently used in ASR systems. It is a simple yet quite powerful method based on the assumption that the current word depends only on the $n-1$ preceding words. A class-based n-gram LM[4] is another way of avoiding data sparseness by mapping words into classes. A common method to improve the n-gram LM is by combining these two LMs using a linear interpolation. To build these models, we used the HTK LM toolkit[15] and the same smoothing method used in the TDC LM (Katz backoff with absolute discounting). The class-based LM was employed with 2000 classes. The perplexity is given in Table 1. The best perplexity achieved by a conventional n-gram LM is 98.9.

5.1.2 Topic Dependent Class LM

The TDC perplexity for varying numbers of topics and window sizes are given

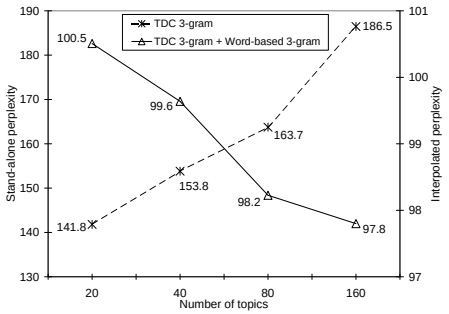

Fig. 1 TDC 3-gram perplexity in increasing number of topics with window size 80.

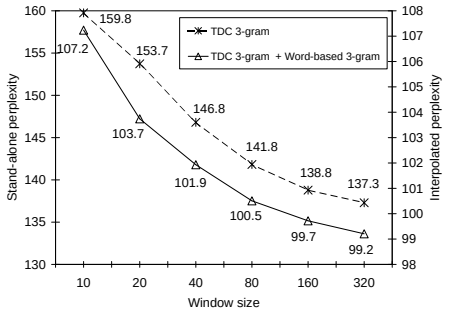

Fig. 2 TDC 3-gram perplexity in increasing window size with 20 topics.

in Fig. 1 and Fig. 2, respectively. Comparisons with each of the baseline methods are discussed in the following subsections. From these results we can see that increasing the window size improves the performance. But when increasing the number of topics, the perplexity in the stand-alone model deteriorates. However, in the interpolated model, the perplexity decreases. This is because in the TDC models, the training corpus is shrunk according to the number of topics. For instance, when the number of topics is 20, each topic is trained with $\frac{1}{20}$ of the corpus on average. This problem can be solved by performing soft-voting on the test phase.

Experiments on soft-voting was made on 1-best (= hard-voting) up to 10-best. Table 2 shows soft-voting based perplexity using 80 topics and 640 window size. With 3-best topics, the performance of TDC 3-gram stand-alone model could overperform the baseline word-based 3-gram. The interpolated model achieved the best perplexity 92.7, that is 3.4% relative improvements against hard-voting (=96.0) and 16.9% relative improvements over the word-based 3-gram LM (=111.6). The stand-alone model keeps improving up to 8-best, while increasing voting to 9-best and 10-best gives no further improvements. This could happen because when using k-best voting, the current word (sequence) probability is estimated using k topic specific LMs. In other words, it is predicted by the model that was trained on a corpus of k times larger than the model with hard-voting (1-best). With soft-voting, the topic decision becomes more reliable and stable.

Table 2 Perplexity of soft-voting TDC 3-gram WSJ corpus (topic=80, window=640).

No	k-best	Stand-alone	Interpolated
1	1	153.4	96.0
2	3	111.5	**92.7**
3	8	**106.6**	94.1
4	10	106.7	94.6

5.1.3 Comparison with Cache-based LM

The cache-based LM is based on the notion that words appearing in a document will increase the probability of appearing again in the same document. Given a history h, the unigram cache-based LM is defined by $P_{CACHE}(w_i) = \frac{f(w_i \in h)}{|h|}$, where $f(w_i \in h)$ denotes how many times w_i occurs in the history h. In this paper, we used a cache-based LM with a fixed word history (window) size. The TDC model is similar to this model in the sense that the TDC model remembers nouns that have been seen before (in the window). Thus the TDC model should be comparable with a cache-based LM with the same window size. The cache-based LM is usually used in conjunction with the word-based n-gram using linear interpolation. To construct the cache-based LM, we used the SRILM toolkit[14]. The window sizes were varied between 10 and 640. For comparison, we employed the (hard-voting) TDC LM with number of topics 20 and 80, and varying window sizes between 10 and 640. The results are shown in Figure 3. According to this figure, the TDC LM clearly outperforms the cache-based LM with respect to perplexity. The performance of the cache-based LM improves up to 100.7 as the

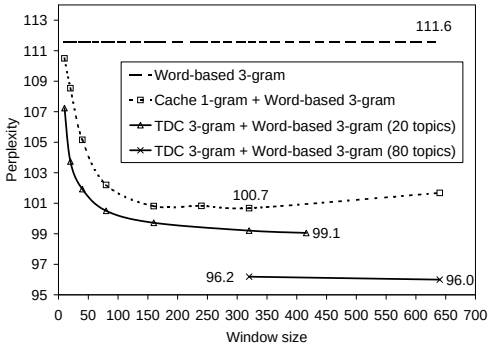

Fig. 3 Cache LM versus TDC LM.

window size varies from 20 to 240, but then deteriorates with the window size between 320 and 640. With regards the TDC LM, the perplexity improves up to 99.1 as the window size increasing to 416. Furthermore, when using 80 topics, the perplexity reached 96.0 with the window size 640.

5.1.4 Comparison with LDA-based Topic Dependent LM

The latent Dirichlet allocation based LM used in this paper is based on reference[11]. This method (referred to as LDA-ADAPT in the remainder of this paper) was proposed to avoid the mismatch problem due to differences in domain, topic, or styles occurring in a general LM. The method splits a large corpus into some topics according to the LDA process by analyzing named-entity words. Then a mixture model is built with mixture weights calculated from analyzing the test corpus in advance. This is a disadvantage if we wish to build a real-time system. An LDA-ADAPT LM with Z topics is defined as follows:

$$P_{LDA-ADAPT}(W) = \prod_{i=1}^{N} \sum_{Z} \gamma_z P_z(w_i | w_{i-1}^{i-n+1}), \tag{7}$$

where mixture weight γ is calculated according to $\gamma_z = \sum_{i=1}^{n} P(z|w_i)P(w_i|d)$, where $P(z|w_i)$ is obtained from LDA analysis. Finally, the LDA-ADAPT is linearly interpolated with the general word-based n-gram LM.

To ensure a fair comparison, we tried to make every parameter as similar as possible. We only considered nouns through a noun-document matrix. The LDA matrix was obtained using the Matlab Topic Modeling Toolbox[⋆1]. We used the HTK LM toolkit because it is able to load more than 9 LMs for the interpolation, unlike the SRILM toolkit which was used in the original paper. We used 20 topics for the comparison, which means 20 topic-specific LMs for the LDA-ADAPT. The window size used in TDC model is equivalent to the LDA-ADAPT context word that is used to capture the long-range constraint, i.e. the average number of words in the test document is 416. The perplexity calculations show that the stand-alone LDA-ADAPT achieved a perplexity of 104.5 while the interpolated LDA-ADAPT achieved 100.6 with $\lambda = 0.6$. Compared with the performance of the TDC model, the stand-alone TDC model, with perplexity 136.8, performs worse than the LDA-ADAPT. With 7-best soft-voting, the TDC (perplexity 105.2) still performs worse than LDA-ADAPT. This might be caused by the pre-analysis of the test dataset of the LDA-ADAPT model. However, the interpolated model, with perplexity 99.1 ($\lambda = 0.4$), outperform the LDA-ADAPT. Futhermore, with 3-best soft-voting, the interpolated TDC perplexity improves to 96.8 ($\lambda = 0.6$). The comparison also conducted on the number of topics 30 and 40 with similar conclusions.

5.2 Mainichi Shimbun Corpus

The training data were taken from the Mainichi Shimbun corpus (Japanese news articles) for the years 1991 to 1998, contains 207,215,663 words in 855,825 documents. Data from the Mainichi Shimbun for January 1999 were used as the test dataset, contains 2,582,469 words in 9,280 documents. Normally, Japanese text does not have spaces between words. For this task, we used the Mecab toolkit[⋆2] (Yet Another Part-of-Speech and Morphological Analyzer), and converted the corpus into basic units, word + part-of-speech. The vocabulary size is 20k words, taken from the most frequent words. With a beginning sentence symbol <s>, an end sentence symbol </s>, and an unknown symbol <UNK> to map all OOV words, the total vocabulary size is $20,001$ words. This gives OOV rate 4.11% and 4.37% for the training and test datasets, respectively. The baseline cache-based LM was executed with an increasing window size from 20 to 640,

⋆1 http://psiexp.ss.uci.edu/research/programs_data/toolbox.htm
⋆2 http://mecab.sourceforge.net

Table 3 Results for Mainichi Shimbun corpus.

No	Model		PP
1	Word-based	3-gram	66.2
2		4-gram	61.4
3	Class-based 3-gram		92.0
4	Class-based 3-gram + Word-based 3-gram ($\lambda = 0.1$)		64.8
5	Cache + Word-based 3-gram (window=160, $\lambda = 0.07$)		59.7
6	Known-topic word-based 3-gram (topic=17)		75.6
7	Known-topic word-based 3-gram + Word-based 3-gram (topic=17, $\lambda = 0.5$)		60.3
8	TDC 3-gram (topic=17, window=320)		70.5
9	TDC 3-gram + Word-based 3-gram (topic=17, window=320, $\lambda = 0.5$)		58.6
10	TDC 3-gram (topic=80, window=320, k=7)		57.2
11	TDC 3-gram + Word-based 3-gram (topic=80, window=320, k=4, $\lambda = 0.5$)		53.6

and the best perplexity was achieved with a window size of 160. Since Mainichi Shimbun corpus contains manually tagged topic information, we also conduct a topic dependent language model based on n-gram (known-topic word-based n-gram). All model's perplexities are given in Table 3. Of all the baseline methods, it can be seen that our model gives the best perplexity 53.6 with $\lambda = 0.7$. This is a relative improvement of about 19.0% on the word-based 3-gram LM.

6. Conclusions and Future Works

We have demonstrated the superiority of the TDC LM over several baseline methods, namely the statistical word-based/class-based n-gram LM and the cache-based LM. The TDC also performed better than the LDA-based topic dependent LM. The TDC LM achieved a relative perplexity improvement over the word-based 3-gram of 14.0% and 15.6% for the WSJ and Mainichi Shimbun corpora, respectively. Soft-voting in the test phase gives even more improvements, up to 16.9% and 19.0% relative for each corpora.

For future works, we will investigate soft-decision on voting in training phase. A soft-clustering on defining topics is also possible to be explored. Then soft-clustering and soft-voting could be combined in several ways in training or test phase. Adding a cache capability in TDC also might improve the performance[9]. Note that in this research, the observation window starts from the outer context of n-gram to avoid information overlap. We will compare this with TDC model where the observation window starts from the immediate word history. Finally, it would be interesting to combine all the methods proposed in this research. Although some models might capture the same aspect of the language.

Acknowledgments This research was supported in part by the Global COE Program "Frontiers of Intelligent Sensing" from the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

1) Bellegarda, J.R.: Latent Semantic Mapping, *IEEE Signal Processing Magazine*, pp.70–80 (2005).
2) Bilmes, J.A. and Kirchhoff, K.: Factored language models and generalized parallel backoff, *in Proceedings of HLT/NACCL, 2003*, pp.4–6 (2003).
3) Blei, D.M., Ng, A.Y., Jordan, M.I. and Lafferty, J.: Latent dirichlet allocation, *Journal of Machine Learning Research*, Vol.3, pp.993–1022 (2003).
4) Brown, P., Pietra, V., deSouza, P., Lai, J. and Mercer, R.: Class-based n-gram models of natural language, *Computational Linguistics*, Vol.18, pp.467–479 (1992).
5) Dhillon, I.S., Fan, J. and Guan, Y.: Efficient Clustering of Very Large Document Collections, *Data Mining for Scientific and Engineering Applications* (R.Grossman, C.Kamath, V.K. and Namburu, R., eds.), Kluwer Academic Publishers, pp.357–381 (2001). Invited book chapter.
6) Dhillon, I.S., Guan, Y. and Kogan, J.: Iterative Clustering of High Dimensional Text Data Augmented by Local Search, *Proceedings of the 2002 IEEE International Conference on Data Mining* (2002).
7) Hofmann, T.: Probabilistic latent semantic analysis, *In Proc. of Uncertainty in Artificial Intelligence, UAI'99*, pp.289–296 (1999).
8) Iyer, R.M. and Ostendorf, M.: Modeling long distance dependencies in language: Topic mixtures versus dynamic cache model, *IEEE Transactions on Speech and Audio Processing*, Vol.7, No.1, pp.236–239 (1999).
9) Kneser, R. and Ney, H.: Improved Clustering Techniques for Class-Based Statistical Language Modelling, *Proceedings of the European Conference on Speech Communication and Technology*, pp.973–976 (1993).
10) Kuhn, R. and de Mori, R.: A cache based natural language model for speech recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, No. 14, pp. 570–583 (1992).
11) Liu, Y. and Liu, F.: Unsupervised Language Model Adaptation via Topic Modeling Based on Named Entity Hypotheses, *ICASSP 2008*, pp.4921–4924 (2008).
12) Rosenfeld, R.: A maximum entropy approach to additive statistical language modeling, *Computer Speech and Language* (1996).
13) Schmid, H.: Probabilistic Part-of-Speech Tagging Using Decision Trees, *Proceedings of International Conference on New Methods in Language Processing* (1994).
14) Stolcke, A.: SRILM – an extensible language modeling toolkit, *Proceedings of ICSLP*, Vol.2, pp.901–904 (2002).
15) Yung, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V. and Woodland, P.: *The HTK book (for HTK version 3.3)*, Cambridge (2005).