Semi-Supervised Learning of Domain-Specific Language Models from General Domain Data

Shuanhu Bai, Min Zhang, Haizhou LI
Institute for Infocomm Research, 1 Fusionopolis Way, #21-01 Connexis, Singapore 138632
{sbai, mzhang, hli}@i2r.a-star.edu.sg

Abstract

We present a semi-supervised learning method for building domain-specific language models (LM) from general-domain data. This method is aimed to use small amount of domain-specific data as seeds to tap domain-specific resources residing in larger amount of general-domain data with the help of topic modeling technologies. The proposed algorithm first performs topic decomposition *(TD) on the combined dataset of domain-specific and general-domain data using probabilistic latent semantic analysis (PLSA). Then it derives domain-specific word n-gram counts with mixture modeling scheme of PLSA. Finally, it uses traditional n-gram modeling approach to construct domain-specific LMs from the domain-specific word n-gram counts. Experimental results show that this approach can outperform both stat-of-the-art methods and the simulated supervised learning method with our data sets. In particular, the semi-supervised learning method can achieve better performance even with very small amount of domain-specific data.*

1. Introduction

LMs are widely used in various natural language processing (NLP) applications such as text mining, machine translation and speech recognition systems. AS we know that the performances of most of the NLP systems are highly domain-dependent. Similar to this nature, LMs, built from the statistical facts of sampled texts, can only achieve better performance when the application domain matches the domain of the training texts. Traditional approach can alleviates this problem by adapting a background model with the data collected from the similar application domains, which is called domain adaptation. In order to achieve better performance, domain specific application systems usually require LMs to be built fully with domain-specific data for a very particular task. Existing learning algorithms for building these LMs rely heavily on the availability of high-quality domain-specific data. Collecting enough such data in most cases is not an easy task, and it is much more difficult to achieve when we are presented with a new domain. The manual approach for data collection is time consuming and expensive. On the other hand, we may have large amount of general-domain data on hand. Tapping into such cheaper resources to alleviate the shortage of domain-specific data seems to be a good choice.

The efforts for building domain-specific LMs have been mostly spent on the issue of obtaining training texts from sources such as the Web[9, 10,12]. Although much of unnecessary data can be filtered out by search engines, the data collected from the Web is still far from direct use. Some semi-supervised methods are employed to identify useful sentences using selection criterions such as BiLingual Evaluation Understudy (BLEU) [9] and relative entropy [10]. Text data selection schemes can be regarded as text/sentence classification methods. Texts or sentences falling into the domain class are used for LM training. In a similar effort, [7] created cluster-based LMs for clustering-based retrieval using clustered corpus.

In recent yes, topic modeling methods have been introduced into language modeling efforts for unsupervised topic adaptation[2, 3, 6, 11]. Although these topic-model-based adaptation methods can not be used for generating high performance domain-specific LMs, their basic idea of using latent topic as means to tap into domain-specific knowledge is heuristic for our semi-supervised learning. On the other hand, semi-supervised learning techniques have been successfully used in other NLP tasks. Typical instances of such applications are the efforts for learning text classification systems using small number of labeled texts and larger pools of unlabeled texts [1, 8, 13], and the resulting models show significant performance improvements.

In this paper, we focus the problem of learning domain-specific LMs from less domain-specific data in a semi-supervised manner. Given a small domain-specific dataset D^I, we are going to build a domain-specific LM by taping into available larger general-domain dataset D^G for "useful" information. Our goal

978-0-7695-3904-1/09 $26.00 © 2009 IEEE
DOI 10.1109/IALP.2009.65

is to utilize this information in D^G to composite for the insufficiency of domain data D^I to build higher performance LMs. In order to achieve this goal, a novel learning approach is proposed.

The motivation behind this approach is based on the observation that text documents in similar domains are inclined to share similar topics; language uses such as words and phrases are highly associated with topics. By tapping into the shared topics of documents of both domain-specific and general-domain data, we can obtain shared language uses such as word co-occurrences and n-gram sequences.

The key idea of our approach is to use latent topics of a topic model as a means of learning to obtain domain-specific language use knowledge from general-domain data. We can make use of the TD mechanisms of a topic model to derive document-dependent topic distribution of the documents in the training sets. Their topic distributions may appear similarity to some extent if documents from D^I and D^G share some similar topics. Thus the topic distributions of the documents actually provide a bridge between dataset D^I and D^G. It allows us to derive language uses such as n-grams in D^G that are highly associated with topics that D^I prefers. Based on the TD result and the assumptions above, we can derive topic-specific word n-gram knowledge from the entire dataset and topic distribution of interested domain represented by D^I. Domain specific n-gram data can further be obtained from topic-specific data and topic-mixture modeling scheme of PLSA. Finally, domain-specific LMs are obtained form domain-specific n-gram data.

The major advantage of this learning approach is that it is able to yield high performance LMs with very small amount of domain-specific data. Different from the prior arts, our proposed method does not perform text selection. It is concerned with selecting fine-grained units of n-grams.

The rest of the paper is organized as follows: Section 2 will be dedicated to detailed introduction to the modeling methods as well as the learning algorithm. We show the experimental results in section 3, and conclude our paper with section 4.

2. Learning method

2.1. Learning Strategies

Under the framework of PLSA[2, 4], word w distribution in document d can be described as mixture of latent topics t:

$$p(w \mid d) = \sum_t p(t \mid d) p(w \mid t) \qquad (1)$$

where $p(t|d)$ represents topic distribution of the documents, while $p(w|t)$ represents mixture components in the form of word unigram model. Both $p(t|d)$ and $p(w|t)$ can be obtained by applying Expectation Maximization (EM) algorithm on the likelihood of a document collection. The training process is referred to as *topic decomposition* (TD). Suppose we have a combined dataset $D = D^G \bigcup D^I$, we treat D^I not only as a set of documents, but as a domain as well. After applying TD on data D, we can approximate latent topic distribution of D^I by treating D^I as a single document, which can be expressed as:

$$p(t \mid D^I) = \frac{\sum_{w,d^I} n(d^I, w) p(t \mid d^I, w)}{\sum_{w,d^I,t'} n(d^I, w) p(t' \mid d^I, w)} \qquad (2)$$

where d^I represents the elements of D^I and $n(d^I, w)$ is the count of word w in document d^I. In PLSA $p(t \mid d^I, w)$ is interpreted as the probability of topic t is used by document d^I to generate word w, thus term $n(d^I, w) p(t \mid d^I, w)$ is the number of times topic t is used by d^I for generating w. We can use the E step of TD method [4] to evaluation of $p(t \mid d^I, w)$. Topic distribution $p(t \mid D^I)$ can be regarded as the latent topic preference of the domain represented by D^I. Since the topic-specific LMs will be working in the same domain as D^I represents, we assume that the topic distribution of incoming documents could be simply modeled with $p(t \mid D^I)$ during decoding. Therefore, the domain-specific LM can be expressed as:

$$p^I(w \mid h) = \sum_t p(t \mid D^I) p(w \mid h, t) \qquad (3)$$

where $p(w \mid h, t)$ are topic-specific n-gram models. It is more reasonable to assume that $p(w \mid h, t)$ are word unigram models because that is the assumption of PLSA. Latent topics here only serve as intermediate variables in building the domain-specific models and are summed out afterwards. Now the problem becomes the issue of TD and derivation of high-order n-gram models $p(w \mid h, t)$.

2.2 Weighted Topic Decomposition

We know that the parameters of PLSA topic models are estimated from the entire dataset through EM. As it is assumed at the very beginning, we may have only a few domain-specific texts available, and there are plenty general-domain texts on hand. If we feed the combined dataset into the training process indiscriminately, the determining force for parameter estimation will be dominated by the general-domain data and the effect of domain-specific data may be ignored. The solution for such problems is to use multi-conditional learning scheme[1] where a weighted objective function can be used, which can be specified as:

$$O(\Theta) = P(D^I;\Theta)P(D^G;\Theta)^\lambda \qquad (4)$$

where $P(D^I;\Theta)$ and $P(D^G;\Theta)$ represent the likelihoods of domain-specific data D^I and general-domain data D^G. A new parameter λ introduced into the likelihood function, can decrease the contribution of the general-domain data during parameter estimation when we choose $0 \le \lambda \le 1$. In practice, it is convenient to maximize the log-likelihood of O :

$$\log O(\Theta) = \log P(D^I;\Theta) + \lambda \log P(D^G;\Theta) \qquad (5)$$

It is obvious that the different likelihoods of D^I and D^G share the same set of parameters. The learning objective is to choose the model parameters $\hat{\Theta}$ that maximize the log likelihood. When we apply this learning strategy to PLSA framework specified by Eq.(1), the log-likelihood of general-domain data in Eq.(5) can be expanded as:

$$\log P(D^G:\Theta)^\lambda = \sum_w \sum_{d^G} \lambda n(d^G,w) \log p(w \mid d^G) \qquad (6)$$

where $d^G \in D^G$, $n(d^G,w)$ is the count of word w in document d^G and $p(t \mid d^G)$ can further be expanded by Eq.(1). Most importantly, we notice that parameter λ is always coupled to the counts $n(d^G,w)$ in the log-likelihood function, therefore it can be regarded as a weighting factor for the document-word counts of the general-domain data. We can use a revised EM algorithm for PLSA model training, which is very much similar to EM$-\lambda$ used in [8].

2.3. Weighted N-gram Counts

PLSA topic model is known as a mixture model and its mixture components are word unigram models. Our objective is to derive high-order mixture components for better performances.

A direct solution for building high-order mixture components is to derive topic-specific n-gram counts first. Then the topic-specific n-gram models can be constructed with conventional n-gram modeling method from the counts. For the convenience of later discussion, we use hw to represent a word n-gram sequence. Here h stands for a word sequence of length $n-1$, it becomes an empty string when word unigram is represented, w is an arbitrary word. Given a document set D and $d \in D$, if we take the view that $p(t|d)$ is the result of *soft classification* of the documents in D , then the count of n-gram hw in the training corpus with respect to latent topic t , by taking the weighting factor λ for datasets D^G and D^I into consideration, can be express as:

$$n(hw,t) = \sum_d \delta(d) n(d,hw) p(t \mid d) \qquad (7)$$

where $n(d,hw)$ is the original count of n-gram hw in document d , $\delta(d)$ will be the weighting factor λ whenever d is in D^G, and will be 1 whenever d is in D^I.

Topic-specific unigram probabilities $p(w|t)$, obtained from DT, may not be used directly in language modeling tasks because stop-words have been removed from the vocabulary before TD. But these words are usually higher frequency words and play important roles, hence must be included in our component models.

It is obvious that topic-specific word unigram counts can be obtained with Eq.(7) if we stick to the soft classification paradigm. As $p(w|t)$ got from TD are considered to be better optimized, thus we use PLSA topic modeling assumptions to derive topic-specific word unigram counts $n(w,t)$ with

$$n(w,t) = \sum_d \delta(d) n(d,w) p(t \mid d,w) \qquad (8)$$

where $p(t|d,w)$ is the probability that topic t is used by document d for generating word w, it follows probability normalization rule. If w is used as a stop-word during TD, we use $p(w|t) = p(w)$ as we assume that these words are topic independent. Eq.(8) can be regarded as a re-normalization process for $p(w|t)$ of TD after the vocabulary is changed. Eq.(7) and Eq.(8) can be regarded as a weighting process for n-gram counts $n(d,hw)$ by topic distribution.

After topic-specific n-gram counts have been derived, we can estimate topic-specific n-gram model parameters with maximum likelihood approach, which can be expressed as:

$$p(w \mid h,t) = n(hw,t) / n(h\cdot,t) \qquad (9)$$

where $n(h\cdot,t)$ represents the total count of the word sequence h followed by any word. By using Eq.(3) and Eq.(9), we can further specify the domain-specific model as follows:

$$\begin{aligned} p^I(w \mid h) &= \sum_t p(t \mid D^I) n(hw,t) / n(h\cdot,t) \\ &= \frac{1}{n(h\cdot,k)} \sum_t \frac{p(t \mid D^I) n(h\cdot,k)}{n(h\cdot,t)} n(hw,t) \qquad (10) \\ &= \frac{1}{n(h\cdot,k)} \sum_t \alpha(h,t) n(hw,t) \end{aligned}$$

Here k theoretically can be any of t. For the convenience of later smoothing, we let $n(h\cdot,k)$ be the mean of $n(h\cdot,t)$ to obtain domain-specific n-gram counts $\sum_t \alpha(h,t) n(hw,t)$, which can be regarded as the mixture of topic-specific n-gram counts $n(hw,t)$ with mixture weights:

$$\alpha(h,t) = p(t \mid D^I) n(h\cdot,k) / n(h\cdot,t) \qquad (11)$$

Therefore, the modeling effort is changed from *mixture of probabilities* to *mixture of counts*. That is, instead of estimating the probability parameters for each of the component models, we can conduct count merging first. Thus we can save the smoothing efforts for each individual topic-specific model. This process can also be regarded as an n-gram weighting scheme[5] using topic distribution of documents and topic distribution of domain. Afterwards, the domain-specific models can be built from the final counts by applying smoothing methods such as cut-off and back-off technologies.

3. Experiments

3.1. Datasets

Our experiments are carried out with part of LDC corpus NA_News98 and some data from 20Newsgroups(http://people.csail.mit.edu/jrennie/20Newsgroups/ , version 20news_bydate). Table 1 presents the structure of the datasets used in our experiments. Because texts of NA_News98 are well categorized into different domains, it enables us to conduct simulation experiments using comparatively larger scale datasets. Dataset D^G is constructed by randomly choosing texts from subset NYT/1997, it consists of total 106,431 documents, and 13,239 of them are in the category of 's' (sports). We compile two groups of D^I , one is referred to as *easy domain* where the documents are selected from subset NYT/98 with the same category 's'. We call it *easy domain* because D^G does contain documents of the same domain as D^I does. In order to study the relationship between the amount of domain-specific data and performance of our learning algorithm, D^Is in this group are created in different size. We also compiled a D^I of a *hard domain* form 20Newsgroups of the category 'sci.med' standing for medical domain. We call it *hard domain* because D^G hardly contains documents of the same domain as D^I does. Besides, the written styles are different. Texts selected from 20Newsgroup for both training and testing have been experienced cleaning up. Given datasets D^G and D^I, document-word tables for TD are built by applying a stop-word list around 500 entries and words with original counts of less than 3 are not used. The vocabulary of the most frequent 60K words for each model is selected from the weighted counts when semi-supervised learning method is used. We use back-off word tri-gram models in our experiments unless it is specially mentioned.

Data	Source	# of Docs
D^G	NA_NEWS98 part of NYT/1997	Total: 106,431 cat='s': 13,239
Easy D^I	NA_NEWS98 part of NYT/1998 with cat='s'	100 / 500 / 900 / 1300 / 1700
Hard D^I	20NEWSGROUP in category 'sci.med'	400
Easy D^I test set	NA_NEWS98 part of NYT/1998 with cat='s'	500
Hard D^I test set	20NEWSGROUP in category 'sci.med'	100

Table 1. Data sets used in experiments

3.2. Experiments on Soft Classification

Our high-order mixture components are constructed based on PLSA soft classification assumption. In order to investigate the soundness of this assumption, we compare the perplexity test results of domain-specific unigram models built with soft classification method indicated by Eq.(8), with that of domain-specific model built with topic modeling method indicated by Eq.(9). We call the later *smoothed topic model* when the parameters of the mixture components are re-estimated after stop-words are folded in and low frequent word are removed from the vocabulary. We also compile the perplexity numbers calculated using a basic unigram model, which is simply a unigram model built from $D^G \cup D^I$ with traditional approach, for comparison purpose. Table 2 shows the test results of the models built from D^G and D^I of different size respectively. The perplexity numbers are obtained applying the test set against *smoothed topic models* and models built with soft classification method. Both kind of models are trained with the setting of 8 latent topics and $\lambda = 1$.

$\|D^I\|$	100	500	900	1300
Basic 1-gram model	1810	1808	1801	1794
Smoothed topic model	1394	1365	1321	1292
Soft Class 1-gram model	1426	1397	1366	1347

Table 2: Perplexity test result of unigram models

We observe that soft classification models perform slightly worse than smoothed topic models do. The performance of the both methods is much better than that of the basic unigram model. Thus, we believe soft classification paradigm is a good n-gram weighting scheme, it can be used to derive high-order mixture components.

3.3. Experiments on Easy Domain

To investigate whether the learning algorithm can effectively tap into the domain knowledge residing in general-domain data, we create domain-specific document sets $D^I s$ in the domain of *sports* in different size, as is indicated by Easy D^I in Table 1. For comparison purpose we build different models from both D^G and $D^I s$ using different approaches:

A. *Supervised learning*. Each model is built from a D^I as well as the documents of category 's' in D^G. It is simply used to simulate manual data collection process.

B. *Domain adaptation.* Each model is constructed by liner interpolation of the background model built form D^G and a domain-specific model built from a D^I.

C. *Relative entropy.* Each model is created with relative entropy text selection scheme[10] which extract relevant documents from D^G with a bootstrap model built from a D^I.

D. *Semi-supervise learning.* Each model is built with our approach as is described in section 2, with the setting of 8 latent topics and optimized λ.

Models created with A, B, C are served as baseline models, models created with D represent our test models. Figure 1 shows the perplexity test results. From Figure 1 we can see that our semi-supervised learning method can easily outperform other three approaches in terms of test set perplexity measure. In the context of our experiments, domain adaptation approach performs the worst. This means that domain adaptation is not a good way to create domain-specific models. The performance of *relative entropy* text selection criteria is between that of *supervised learning* and domain adaptation. We found that it is still very important for this method to have a bootstrap model that is built from sizeable balanced data.

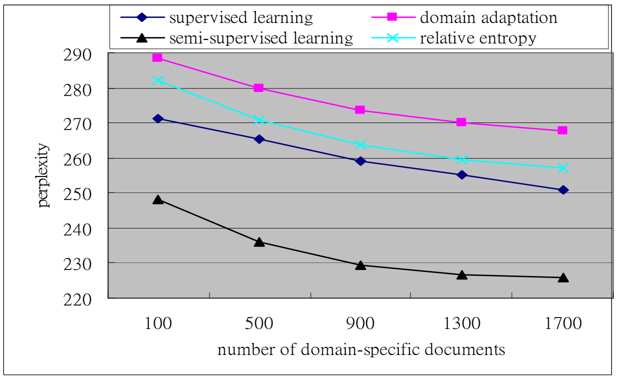

Figure 1: Perplexity test results of different learning approaches with different size of D^I.

We also notice that our semi-supervised learning method outperforms supervised learning method with present configuration. The reason is manifold. First of all, the documents in training set probably are not well categorized, maybe there exist some texts that should be classified as category *sport,* our learning algorithm can manage to find out and make use of them. The simulated *supervised learning* method, on the other hand, is not able to retrieve such texts using simple string matching mechanism and the collected training data is not enough. Another reason may be due to the fact that the category boundaries of texts from news media are not so clear. Our algorithm may somehow *borrow* some information, such as topic-specific word co-occurrence preferences that an n-gram model wants to capture, from texts of other domains.

In addition, our semi-supervised learning algorithm works well with smaller domain-specific dataset. Particularly, with the D^I of only 100 texts, which is only about 0.1% of the size of D^G, we can yield models with higher performances than those created by other three methods. Its perplexity of 248.3 at this point is the lowest comparing with 271.4 of the second best performing method of *supervised learning* at the same point and 251 at 1,700. At the same time, performance improvement with D^I of size 1,700 over D^I of size 1,300 is not distinctive.

3.3 Experiments on Hard Domain

Experiments on easy domain show that our learning method is able to learn domain specific knowledge form general-domain data for easy domains. It is much more important if it is able to learn knowledge from general-domain data for hard domains. In order to carry out such experiment, we create a domain-specific dataset hard D^I as well as a hard test set, as are shown in Table 1. The experimental results are presented in Table 3.

Modeling methods	Perplexity	OOV rate
Baseline built from D^G	533	5.4%
Baseline(D^G)+Domain(D^I)	359	3.3%
SSL with T=12, λ =1.0	317	3.6%
SSL with T=12, λ =0.8	281	2.8%
SSL with T=12, λ =0.6	263	2.1%
SSL with T=12, λ =0.4	292	3.4%

Table 3: Experimental results on hard domain.

We observe form Table 3 that there exists significant domain mismatch between D^G and the test set. Without the help of D^I, the baseline model built from D^G alone generate an astonishing perplexity of 533 and out-of-vocabulary (OOV) rate of 5.4% against the test set. Domain adaptation with liner interpolation method can achieve significant perplexity reduction even with a very small amount of data. But our semi-supervised learning (SSL) method can make further improvement. Contrary to the phenomena in previous experiment, our method can only reach optimal state by setting λ with smaller value of 0.6. This can be explained by the fact that D^G does not clearly contain much domain-specific data, we need to decrease its influence over D^I to find useful information during DT and n-gram model training. We also notice that both perplexity number and OOV rate are consistently improving when we decrease λ from 1 to 0.6. This means that the learning algorithm is able to extract more helpful information from the training corpus through appropriately re-weighting of the datasets.

3.4 Experiments on Parameter Setting

As mentioned earlier, there are two free parameters need to be set for our learning algorithm: the number of latent topics T and the weighting factor λ for general-domain data. T will directly affect DT performance while λ will be involved in both DT and n-gram model parameter estimation. In order to investigate how these parameters affect the learning performance, we conduct experiments on different parameter setting.

Figure 2 shows the performance of the models obtained with different weighting factor λ and 8 latent topics. The experiments are carried out with D^I in different size. We observe that our learning algorithm can hardly achieve best performance with the setting of $\lambda = 1$. Therefore the weighting factor is important for us to derive higher-performance models. In particular, model obtained with D^I of size 100 become little bit worse off with perplexity of 249 when λ is set 0.8 comparing perplexity of 248.3 when λ is set 1. On the other hand, the overall trend becomes worse when λ is set with 0.4 or below. This can be explained by the fact that excessively lowering the weighting for D^G equals to intensively raising the weighting for D^I. The effects of D^I and its closely related documents in D^G to some extent can also been factored into the models through topic-specific n-gram counts as well as $p(t \mid D^I)$.

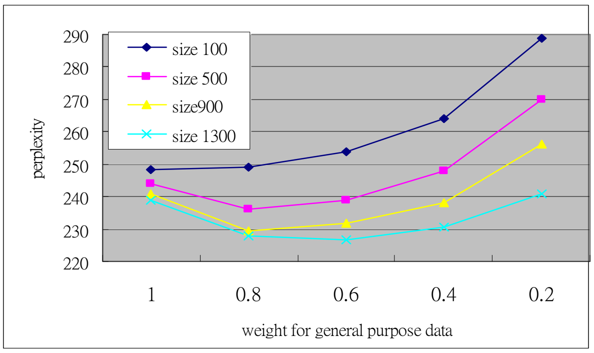

Figure 2: Perplexity test results of different models obtained with different size of D^I and different λ .

We also study the effect of number of latent topics on the performance of our learning algorithm. Similar studies have been made in the pioneer work of topic modeling[4] as well as its applications[6,11]. Here we focus on issues that are particular to our learning algorithm: the impact on high-order mixture components and on weighting factor λ. Due to high computation requirement we only study two cases. Table 4 summarizes the experimental results with different number of topics.

# latent topic T	4	8	12	16	20
$\lvert D^I \rvert$=500 , $\lambda = 1$	267	239	229	224	221
$\lvert D^I \rvert$=1300 , $\lambda = 0.8$	261	227	222	219	218

Table 4: Perplexity test results for different number of latent topics.

From Table 4 we notice that larger number of topics can result in better performance, which is in line with the results of prior arts. But the performance does not improve much when T reaches 16. This trend is different from the results of [6] and [11] where much larger numbers of topics are applied (from 50 to 200 topics) with word unigram models. The experiments also preliminarily reveal that the setting of λ has no direct impact on the setting of T .

4. Conclusions

In this paper we proposed a novel semi-supervised learning method for building domain-specific LMs. The innovative aspects of our method are: the learning strategy and the derivation of topic distribution of interested domain; the weighted TD method for combined dataset of domain-specific and general-domain data; N-gram weighting strategies for domain-specific models. The whole learning process is under the multi-conditional learning scheme which can effectively balance the influence of the domain-specific and general-domain data. We conducted experiments on easy domain as well as hard domain and the results show that the proposed method is very effective. It can not only achieve better performance than state-of-art semi-supervised learning method that uses relative entropy as text selection criteria, it can also deliver better result than the simulated supervised learning process does with the present configuration.

As future works, we may extend the learning strategy to other domains. We will also consider using other topic modeling method to make the learning method more effective.

References

[1] Druck, G., Pal, C., Zhu, X., McCallum, A., "Semi-Supervised Classification with Hybrid Generative/ Discriminative Method". KDD'07. August 12-25, CA USA, 2007.

[2] Gildea, D. and Hofmann, T., "Topic-based language models using EM", Proc. of Eurospeech. 1999.

[3] Heidel, A., Chang, H.A. and Lee, L.S., "Language Model Adaptation Using Latent Dirichlet Allocation and Efficient topic Inference Algorithm", INTERSPEECH'2007.

[4] Hofmann, T., "Unsupervised Learning by Probabilistic Latent Semantic Analysis", Machine Learning, 42,177-196,2001.

[5] Hsu, B. J., and Glass, J., "N-gram Weighting: Reducing Training Data Mismatch in Cross-Domain Language Model Estimation", p829-838, Proc. EMNLP'08, 2008.

[6] Liu, F. and Liu, Y., "Unsupervised Language Model Adaptation Incorporating Named Entity Information", ACL'2007, Prague, Czech Republic. 2007.

[7] Liu, X., and Croft, W.B., "Cluster-Based Retrieval Using Language Model" SIGIR'04, July 25-29, UK, 2004.

[8] Nigam, K., McCallum, A.K., Thrun, S., and Mitchell, T.M., "Text classification from labeled and unlabeled documents using EM", machine learning , 39, 103-134, 2000.

[9] Sarikaya, R., Gravano, A. and Gao, Y., "Rapid language model development using external resources for new spoken dialogue domain", ICASSP2005, 2005.

[10] Sethy, A., Georgiou, P.G., and Narayanan, S., "Text data acquisition for domain-specific language models" p382-389, EMNLP 2006.

[11] Tam, Y. and Schultz, T., "Dynamic Language Model Adaptation using Variational Bayes Inference", INTERSPEECH'05, 2005.

[12] Wan, V., Hain, T., "strategies for language model web-data collection", ICASSP'2006, 2006.

[13] Xue, G.R., Dai, W.Y., Yang, Q.and Yi, Y., "Topic-bridged PLSA for cross-domain text classification", SIGIR'08 July20-24, 2008, Singapore.