A Latent Variable Model for Query Expansion Using the Hidden Markov Model

Qiang Huang
Knowledge Media Institute
The Open University
Milton Keynes, UK
q.huang@open.ac.uk

Dawei Song
Knowledge Media Institute
The Open University
Milton Keynes, UK
d.song@open.ac.uk

ABSTRACT

We propose a novel probabilistic method based on the Hidden Markov Model (HMM) to learn the structure of a Latent Variable Model (LVM) for query language modeling. In the proposed LVM, the combinations of query terms are viewed as the latent variables and the segmented chunks from the feedback documents are used as the observations given these latent variables. Our extensive experiments shows that our method significantly outperforms a number of strong baselines in terms of both effectiveness and robustness.

1. INTRODUCTION

In information retrieval, a prominent query language modeling approach is the Relevance Model (RM) [3]. Practically, variants of RM have shown encouraging performance in adhoc search [3, 4, 1] and cross-language retrieval [2], etc.

However, in these existing models, three issues, which in our belief will affect the retrieval effectiveness significantly, have not been addressed. Firstly, there often exist dependencies between query terms (i.e., intra-query term dependencies). Secondly, the distributions $P(M_j)$, i.e., the importance of different relevance feedback documents, should not be kept uniform. Instead, they should depend on the query terms and w. Thirdly, noise often exists within the feedback documents not any part of a relevant or pseudo-relevant document is necessarily relevant to the query.

The hypothesis of this paper is that tackling all the above identified three issues in a single query language modeling framework will further improve the retrieval effectiveness. In this paper, we first propose segmenting a document into chunks by using an overlapped sliding window. We decompose the query into exhaustive combinations (subsets) of query terms and consider them as latent variables over the chunks. We then propose using a Latent Variable Model (LVM) which connects a chunk d and a word w through the latent variables. The dependencies between the latent variables are governed by an ergodic Hidden Markov Model (HMM), where the Viterbi algorithm is applied to optimize parameters involved in the HMM and the underlying LVM.

2. FRAMEWORK OF THEORY

In our model, the combinations of query terms are seen as the latent variables over the top-ranked documents. The reasons for selecting the query terms as the latent variables are that the word dependence to the query Q is to be estimated in the query language modeling framework, and the estimation of co-occurrence of words with query terms is key to query expansion. Eq. 1 shows an intuitive derivation of theory:

$$P(w|\theta_Q) = \sum_{S_j \in \mathbf{S}} P(w|S_j)P(S_j) \tag{1}$$

S_j is a latent variable in the set $\mathbf{S}$ ($\mathbf{S} = \{S_1, \cdots, S_M\}$) and w is the word whose occurrence probability in the expanded query model θ_Q to be estimated. The latent variable S_j is generated from the query Q, where each S_j is defined as a combination of query terms. For example, given $Q = \{q_1, q_2\}$, the set of latent variables can be represented as $\mathbf{S} = \{\{q_1\}, \{q_2\}, \{q_1, q_2\}\}$. Compared with the traditional methods the use of all the combinations of query terms expands the observing space.

In Eq. 1, the relationship between the word w and the latent variable S_j is derived from the relevance feedback documents. As we have introduced in Section 1, the top-ranked documents in pseudo relevance feedback are not necessarily relevant to the query. Thus, we propose using the segmented chunks to connect S_j and w, and the relevance of each chunk to the query will also be considered. Then, a new equation is given as below:

$$P(w|\theta_Q) = \sum_{d_i \in \mathbf{d}, S_j \in \mathbf{S}} P(w|d_i, S_j)P(d_i|S_j)P(S_j) \tag{2}$$

$\mathbf{d}$ is the collection of chunks. $P(S_j)$ is the prior distribution of latent variables, $P(d_i|S_j)$ is the probability of an observed chunk d_i given a latent variable S_j, and $P(w|d_i, S_j)$ is the probability of a word w in a chunk d_i given a specific latent variable S_j.

In order to calculate the three probability parameters in Eq. 2, we design a framework based on the Hidden Markov Model (HMM). The application of the HMM can not only estimate the prior distribution of each S_j, but also integrate the dependence between any two latent variables and their underlying observables through a state transition matrix. The details are presented in Figure 1.

Here, we regularize the model estimation with the original query model $P(q_k|Q)$ to alleviate the data sparsity. In this

1. **Pre-processing**

   1.1 Generate "hidden" states by combining the query terms.

   1.2 Select N top-ranked relevant or pseudo-relevant documents according to the initial retrieval results.

   1.3 Segment the selected document into chunks with an overlapped sliding window, whose overlapping length is 4/5 of the sliding window size.

   1.4 Retain the chunks containing any query terms and discard the rest.

   1.5 Assign those chunks containing query terms into various clusters, labelled by the "hidden" states that share one or more query terms with the chunks.

2. **Optimize hidden Markov model**

   2.1 Set the initial values of the model parameters.

   $$P(S_j) = 1/M, \qquad (M = |\mathbf{S}|, \quad S_j \in \mathbf{S})$$
   $$P(S_{j'}|S_j) = 1/M, \qquad (M = |\mathbf{S}|), \quad S_j, S_{j'} \in \mathbf{S})$$
   $$P(d_i|S_j) = \frac{\sum_{t=1,d_t=d_i}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$$

   where $\gamma_t(j) = P(S_j|d_t, \Lambda)$, and $P(S_j|d_t, \Lambda)$ can be approximately derived by the pseudo code as follows:

   ```
   $d_{i,k} = \{w_1, \cdots, w_k\}, \quad d_i = d_{i,K} \ (K = |d_i|)$
   $P(S_j|d_{i,0}) = P(S_j)$
   $for\ k = 1 : K,$
       $P(S_j|d_{i,k}) = \frac{1}{k+1} \frac{P(w_k|S_j)P(S_j|d_{i,k-1})}{\sum_{S_t} P(w_k|S_t)P(S_t|d_{i,k-1})} + \frac{k}{k+1} P(S_j|d_{i,k-1})$
   $end$
   $P(S_j|d_i) = P(S_j|d_{i,K})$
   ```

   The computation of $P(w_k|S_j)$ is detailed in [5].

   2.2 Apply Viterbi algorithm to searching the optimal state sequences.

   2.3 Collect the labelled chunks of each "state" and update the occurrence probability of the observed term w_k, namely $P(w_k|S_j)$, then $P(d_i|S_j)$.

   2.4 Optimize the model iteratively by repeating Step2.1 ∼ Step2.3.

3. **Derive the language model**

   Compute the final probability of $P_{hmm}(w|\theta_Q)$ using Eq. 2.

Figure 1: Outline of framework

paper, the original query model is defined as below:

$$P(q_k|Q) = \frac{\#q_k \cdot IDF(q_k)}{\sum_{j\in\{1\cdots|Q|\}} \#q_j \cdot IDF(q_j)} \qquad (3)$$

where $\#q_k$ is the frequency of query term q_k in Q and $IDF(q_k)$ is the inverse document frequency (IDF) of q_k.

To mix the original query model into the newly generated one, two methods are used. **HMM-I** is an automatic method to integrate the original query model directly into the HMM, in which the original query model as a hidden state S_Q. We therefore can obtain Equation 4.

$$P_{HMM-I}(w|\theta_Q) = \sum_{d_i\in\mathbf{d}} \sum_{S_j\in\{\mathbf{S},S_Q\}} P(w|d_i, S_j)P(d_i|S_j)P(S_j) \qquad (4)$$

The second method, **HMM-II**, uses the linear interpolation to combine the original query model.

$$P_{HMM-II}(w|\theta_Q) = \lambda P(w|\theta_Q) + (1 - \lambda)P(w|Q) \qquad (5)$$

The latter method involves manual adjustment of the interpolation parameter to generate the optimal retrieval performance.

3. EXPERIMENTS AND CONCLUSION

We evaluate our methods by testing TREC topics (Topics151–200, 601–700, and 501–550)on three large collections (AP88–90, ROBUST, and WT10G), respectively. we apply our approach to two scenarios: **pseudo-relevance feedback** and **true relevance feedback**. As a comparison, we list the the values of MAP obtained by using HMM-II and HMM-I in Table 1.

Table 1: Comparison of Optimal MAPs using HMM-II and HMM-I

Pseudo-relevance Feedback					
Collection	KL	RM1	RM2	HMM-I	HMM-II
AP88–90	0.2077	0.2603	0.2676	0.2814†	0.2830†
ROBUST	0.2920	0.3129	0.3143	0.3613†	0.3660†
WT10G	0.2032	0.2131	0.2134	0.2305†	0.2370†
Relevance Feedback					
Collection	KL	RM1	RM2	HMM-I	HMM-II
AP88–90	0.2077	0.3218	0.3427	0.4034†	0.4168†
ROBUST	0.2920	0.4126	0.4432	0.5014†	0.5122†
WT10G	0.2023	0.2571	0.2993	0.3764†	0.3859†

† The improvement is statistically significant at the level of 0.05 according to the Wilcoxon signed rank test

Table 1 shows the optimal value when using HMM-II. It is found that using HMM-II can only obtain slightly and insignificantly better performances than HMM-I. However, HMM-II is a somehow heuristic approach compared with HMM-I which integrates original query model neatly in the Hidden Markov Model in a fully automatic way. Therefore, the HMM-I has demonstrated its robustness and effectiveness from both theoretical and practical perspectives.

In this paper, we present a novel method to build a latent variable model using the HMM for query expansion. This paper tries to address some specific issues in those existing models. Our technique aims to incorporate these key issues in a single comprehensive framework and apply the HMM to estimating and optimizing the structure of the LVM. Our experimental results therefore show that our method always obtains significant improvements in comparison with KL, RM1 and RM2.

In terms of future work, it would be very interesting to continue the studies on how to measure the quality of feedback documents and the relevant contents in these documents.

4. REFERENCES

[1] J. Bai, D. Song, P. Bruza, J. Nie, and G. Cao. Query expansion using term relationships in language models for information retrieval. In *Proceedins of CIKM'2005.*

[2] V. Laverenko, M. Choquetto, and W. B. Croft. Cross-lingual language models. In *Proceedings of SIGIR'2002.*

[3] V. Laverenko and W. B. Croft. Relevance-based language models. In *Proceedings of SIGIR'2001.*

[4] D. Song and P. Bruza. Towards context sensitive information inference. *JASIST*, 54(3):321–334, 2003.

[5] D. Song, Q. Huang, S. Rueger, and P. Bruza. Facilitating query decomposition in query language modeling by association rule mining using multiple sliding windows. In *Proceedings of ECIR'2008.*