Applying Language Model into IR Task

Zhang Junlin, Sun Le, Zhang Yongchen, Sun Yufang
Chinese Information Processing Center,
Institute of Software,Chinese Academy of Sciences,
P.O.Box 8717,Beijing,100080,P.R.China
E_mail:junlin01@iscas.cn

Abstract

This paper mainly describes the methods and procedures we took in participating in NTCIR4 CLIR track. We focus our experiments on evaluating the effectiveness of the language model based IR method. At the same time, in this paper we also propose the trigger language model based IR system to relax the independent assumption of words within the classical language model IR method. The analysis of the language model based IR method compared with VSM method is also presented in the paper.

Keywords: Information Retrieval, Language Model

1 Introduction

The language modeling approach to information retrieval (IR) is a new framework that has been proposed and developed within the past five years, although its roots in the IR literature go back more than twenty years. Research carried out at a number of sites has confirmed that the language modeling approach is a theoretically attractive and potentially very effective probabilistic framework for building IR systems.[1,2,3,4,5,6,7]

This paper mainly describes the methods and procedures we took in participating in NTCIR4 CLIR track. We focus our experiments on evaluating the effectiveness of the language model based IR method. At the same time, in this paper we also propose the trigger language model based IR system to relax the independent assumption of words within the classical language model IR method.

This paper is organized as follows: Section 2 is the brief introduction of the language model based IR system. Section 3 describes the language model methods and relative procedures we took in Monolingual IR subtasks and Bilingual CLIR subtasks in NTCIR4. In Section 4 we analyze the experiment results. We propose a trigger language model IR system in section 5. Section 6 concludes this paper and previews our future work.

2 Language Model Based IR System

Recent advances in Information Retrieval are based on using Statistical Language Models (SLM) for representing documents and evaluating their relevance to user queries. Language Model (LM) has been explored in many natural language tasks including machine translation and speech recognition. In LM approach to document retrieval, each document D is viewed to have its own language model, M_D. Given a query Q, documents are ranked based on the probability, $P(Q|M_D)$, of their language model generating the query. While the LM approach to information retrieval has been motivated from different perspectives, most experiments have used smoothed unigram language models that assume term independence for estimating document language models.

In most approaches, the computation of language model based IR method is conceptually decomposed into two distinct steps: (1) Estimating a document language model; (2) Computing the query likelihood using the estimated document model based on some query model. For example, Ponte and Croft [1] emphasized the first step, and used several heuristics to smooth the Maximum Likelihood of the document language model, and assumed that the query is generated under a multivariate Bernoulli model. The BBN method [2] emphasized the second step and used a two-state hidden Markov model as the basis for generating queries, which, in effect, is to smooth the MLE with linear interpolation. In Zhai and Lafferty [8], it has been found that the retrieval performance is affected by both the estimation accuracy of document language models and the appropriate modeling of the query, and a two stage smoothing method was suggested to explicitly address these two distinct steps.

Being able to estimate retrieval parameters is a major advantage of using language models for information retrieval. Another advantage of using language models is that we can expect to achieve better retrieval performance through the more accurate estimation of a language model or through the use of a more reasonable language model. Thus, we will have more guidance on how to improve a retrieval model than in a traditional model. Finally, language models are also useful for modeling the sub-topic structure of a document and the redundancy between documents.

3 Our Work at NTCIR4

We participated in 2 subtasks in CLIR track of NTCIR4. Data listed in Table 1 shows the average precision of each subtask we participated in.

In NTCIR4, We aim at evaluating the effectiveness of the Language Model based IR method. So we design several experiments to compare the Language Model IR system with traditional VSM method.

Run Types		Average Precision		
		D	T	DN
C-C	VSM	0.1774	0.1944	0.1774
	Language Model	0.1953	0.1792	/
E-C	VSM	0.0021	0.0273	0.0021
	Language Model	0.0013	0.0184	/

Table1.Average Precision of All Subtasks of ISCAS-------Relax

3.1Monolingual IR Subtask (C-C Run)

Since word boundaries are not marked in Chinese written text, word segmentation is necessary to break Chinese sentences into indexing terms, which can be words, single characters, two characters, and so on. All the subtasks which are relevant with Chinese document collection are word based index. Our segmentation algorithm is called bi-direction maximal match algorithm. It scans the Chinese sentence two times by looking up the maximal match term in a general purpose dictionary: The first time is from left to right and the second time reverse the scan order from right to left. This way we can identify and avoid some type of segmentation ambiguity.

In our experiment in NTCIR4, we performed several different C-C runs based on either VSM or Language Model method to compute the similarity of the query and documents. VSM is employed as a baseline to evaluate the language model method. In VSM, the term of vector is word. If T={ t_j } is a term set, then query vector v_j of topic j can be express

$V_j=(v_{j1},v_{j2},\ldots v_{jn})$,in which v_{jk} denotes the weight of t_k in v_j.The vector $D_i=(d_{i1},d_{i2},\ldots,d_{in})$ denotes a document ,d_{ik} denotes the weight of t_k in d_i.The similarity between v_j and d_i is calculated by following formula

$$s_j = \sum_{k=1}^{n} d_{ik} * v_{jk} \Big/ \sqrt{\sum d_{ik}^2 + \sum v_{jk}^2} \qquad (1)$$

The language model IR method is our main concern in NTCIR4. The query likelihood retrieval method is based on the original language modeling approach proposed by Ponte and Croft in 1998[1]. It involves a two-step scoring procedure. First, estimate a document language model for each document, and, second, compute the query likelihood using the estimated document language model directly. In our experiments, we make use the original language model using the two-stage smoothing approach. These models generalize this two-step procedure by introducing a query generative model. As a result, in the second step, instead of using the estimated document model directly, we use the query generative model, which is based on the estimated document model, to compute the query likelihood.

From the viewpoint of smoothing, we can regard such a two-stage language modeling approach as involving a two-stage smoothing of the original document model. The first-stage smoothing happens when we estimate the document language model, and the second stage is implemented through the query generative model. The two-stage smoothing method can be easily obtained as a special case of the two-stage language models where we use a Bayesian approach to estimate the document language model and a mixture model for query generation.

An important advantage of the two-stage language models is that they explicitly capture the different influences of the query and document collection on smoothing. It is known that the optimal setting of retrieval parameters generally depends on both the document collection and the query; decoupling the influence of the query from that of documents makes it easier to estimate smoothing parameters independently according to different documents and different queries.

3.2 Query Translation of CLIR

The main concern of subtasks in the Bilingual CLIR is query translation. The easiest way to find translations is to look up each query term in a bilingual dictionary. However, We can’t neglect problems brought by this method such as coverage, spelling norms. Applying MT in CLIR is also a straightforward approach. Another option to using translation dictionaries is using a parallel or comparable corpus, that is, the same or similar text written in different languages.

Our aim is the evaluation of the language model IR method in NTCIR4, So we didn’t do much work on the query translation in E-C subtask of NTCIR4. We directly use lexical approach to translate the English query into Chinese. Then we search the relevant documents in the Chinese document collection with our Chinese monolingual IR system.

4 Analysis of the Experiment Results

In order to evaluate the effectiveness of the language model IR method, we design several experiments to compare LM based method with VSM method. Table 2 shows the relationship between the various Run ID and the IR model they adopted.

IR Model	**C-C Run ID**	**E-C Run ID**
Language Model+ 2-stage smoothing	ISCAS-C-C-T-02 ISCAS-C-C-D-04	ISCAS-E-C-T-02 ISCAS-E-C-D-04
VSM	ISCAS-C-C-T-01 ISCAS-C-C-D-03	ISCAS-E-C-T-01 ISCAS-E-C-D-03

Table 2 different runs and their IR model

From the experimental results, we observe the following rules:

(1) In C-C subtasks, we found that the performance of language model method is always better than VSM when we use the “Desc” field as the query (figure 1) while the worse performance is observed when the “Title” field is used as the query (figure 2). Generally speaking, the title fields are the concise and short queries which contain only several keywords compared with the “Desc” field. So we can draw the conclusion from the experiments that language model method can be a better choice if the query is relatively longer. The reason why this happens needs the further investigation.

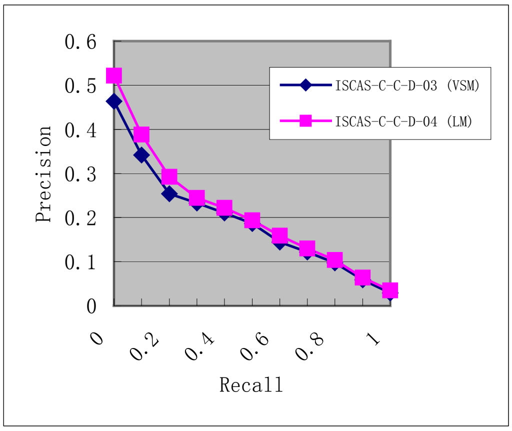

Fig1. Precision-Recall of C-C-D Run

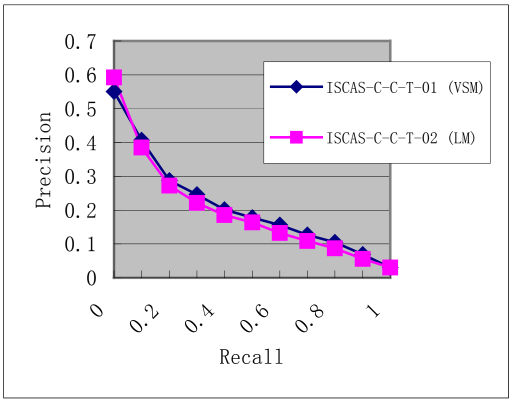

Fig 2. Precision-Recall of C-C-T Run

(2)In E-C subtasks (figure 3), the performance of the language model method is always worse than the VSM, no matter which fields are used as the query (Title or Desc). This result is out of our expectation because we thought the result should be similar with the C-C run. We explain this as following: We only directly use lexical approach to translate the English query into Chinese, so too much noise of translation is introduced into the translated query. The 2-stage smoothing approach need to build the query modeling according to the translated query and the noise in the query play a more important role to build the query modeling compared with the correct translation of query words. While it seems VSM didn’t suffer so much from the worse translation.

(3) In E-C subtasks, we found that the performance get worse whenever we use the “Desc” field as the query compared with the “Title” field. We thought it’s because the “Desc” field is longer compared with “title” field and we just use the simple query translation approach by directly looking up the lexicon. This simple query translation method will bring much noise into the translated query. So much more irrelevant documents are searched out.

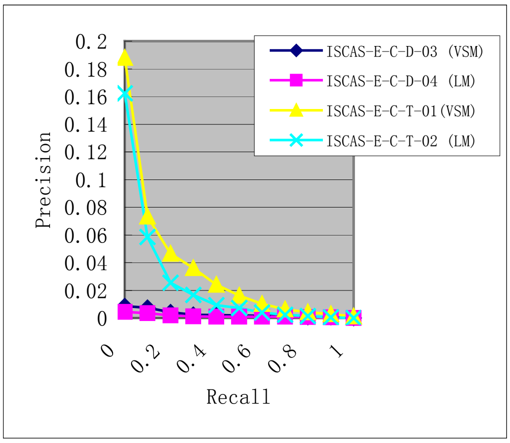

Fig 3. Precision-Recall of E-C Subtask

5.Trigger Language Model for IR

It’s not hard to see that the unigram language model IR method contains the following assumption: Each word appearing in the document set and query has nothing to do with any other word. Obviously this assumption is not true in reality. In this paper we propose the trigger language model based IR system to resolve the problem. Firstly we compute the mutual information of the words from training corpus and then get the triggered words collection of the query words to find the real meaning of the word in specific text context. We introduce the relative parameters into the document language model to form the trigger language mode based IR system.

5.1 Inter-relationship of Indexing Words

In order to find out the inter-relationship of words in some specific context, we consider the co-occur times of different words within fixed sized text window of the document. When the co-occur time is large enough, we think that relationship is meaningful. Mutual Information is a common tool to be applied under this situation. So we compute the mutual information as following:

$$\omega(w_a, w_b) = \frac{\dfrac{N(w_a, w_b, L_w)}{N_w \cdot (L_w - 1)}}{\left(\dfrac{N(w_a)}{N_w}\right) \cdot \left(\dfrac{N(w_b)}{N_w}\right)} \qquad (2)$$

$$= \frac{N(w_a, w_b, L_w) \cdot N_w}{(L_w - 1) \cdot N(w_a) \cdot N(w_b)}$$

where N_w denotes the size of the vocabulary, $N(w_a, w_b, L_w)$ is the co-occur times of word w_a and w_b within L_w sized window in training set. $N(w_a)$ is the times the word w_a appearing in the training set and $N(w_b)$ is the times the word w_b appearing in the training set.

5.2 Algorithm to Find out the Exact Meaning of the Query Words

Generally speaking, a word always represents many different meanings and its exact meaning adopted in specific topic can be determined by the co-occur words in its context. Different meaning of a word often lead to the different vocabulary set of related word.

In order to find out the exact meaning of the words contained by the query in IR system, we design the algorithm to compute the triggered vocabularies of query. It is just these triggered words that show the exact meaning of the words in query in some specific context. The basic idea behind the algorithm is as following: By computing the mutual information, we can derive the relative words of a query word. All these words mean the semantically related vocabularies of the query word under different contexts. We propose that if the intersection of the derived related words of different words in query is not null, the words in the intersection is useful to judge the exact meaning of the words in query. At the same time, the more times an intersection word appears in related vocabulary set of different query word, the higher the weight of this word to fix down the topic of the query is. So we design the following algorithm to compute the triggered vocabulary set of query:

Algorithm 1: Triggered vocabularies by query

Input: Vocabulary set I of query word and its co-occur words after removing the stop words of the query. $I = \{<q_1, S_1>, <q_2, S_2>, ... <q_i, S_i> ...... <q_n, S_n>\}$

Output: Triggered vocabulary set T.

Algorithm:

```
1. Initialize the set $T = \phi$.
2. for(i =2;i<=n;i++)
   {
     for(j=1;j<= $C_n^i$;j++)
     {
       2.1 get the different combination
           $L_j = \{<q_{j,1}, S_{j,1}>, <q_{j,2}, S_{j,2}> ..... <q_{j,i}, S_{j,i}>\}$
           which has $i$ elements from set $I$;
       2.2 if any vocabulary set
           $S_{j,k} (1 < k <= i)$ in $L_j$ contains no
```

```
    element, then we turn to 2.4 , otherwise
    we turn to 2.3;
    2.3 Compute the intersection $T_{i,j}$ of
      all vocabulary set $S_{j,k} (1 < k <= i)$ in $L_j$ .
      here
      $T_{i,j} = \{<w_1,\alpha_1>,<w_2,\alpha_2>,.....<w_m,\alpha_i>\}$
      ,where $\alpha_w = \frac{\log i}{2}$ ,
      ( $1 =< w <= i$ ). $\alpha_w$ is the word
      weight decided by the length of $L_j$ ;
  2.4 $T = T \cup T_{i,j}$ ,adopting the higher
      word weight $\alpha_w$ during the merging
      process;
        }
      }
  3. Output the triggered vocabulary set
$T$ ;
```

5.3 Similarity Computation of Query and Document

We use the similar strategy with classical language model method [1] to compute the similarity between the query and the document. That is, we firstly construct the simple language model according to the statistical information of vocabulary and then compute the generative probability of the query. The difference is that the trigger language model method takes the context information of a word into account. So we compute the triggered words set of query q according to algorithm 1.This way we get the triggered vocabulary set $T_q = \{<w_1,\alpha_1>,<w_2,\alpha_2>,......<w_m,\alpha_m>\}$. This set contains the words triggered by query and it is these triggered words that determine the exact meaning of the vocabularies in query among the several optional choices. Introducing the triggered words factor into the document language model, we can form the trigger language model based information retrieval system.

The similarity of query and document can be computed as following:

$$P(Q \mid M_d) = \prod_{i=1}^{l(Q)} (\sum_{j=1}^{l(d)} C_j \cdot p(q_i \mid d_j) + \frac{tf(q_i)}{cs}) \quad (3)$$

$$p(q_i \mid d_j) = \begin{cases} 1 & q_i = d_j \\ \alpha_j & (q_i \neq d_j) \wedge (d_j \in T_q = \{<w_1,\alpha_1>,<w_2,\alpha_2>,.....<w_m,\alpha_m>\}) \\ 0 & other \end{cases} \quad (4)$$

where ,

(1) $Q = \{q_1, q_2, ... q_i, ... q_{l(Q)}\}$ denotes query and $l(Q)$ is the length of the query;

(2) M_d denotes the trigger language model of document d ;

(3) $d = \{d_1, d_2, ... d_j, ... d_{l(d)}\}$ denotes a document and $l(d)$ is the length of the document;

(4) $C_j = \frac{f(d_j)}{l(d)}$ is the weight parameter of words d_j in a document. here $f(d_j)$ means the account of the words d_j appearing in the document.

(5) $p(q_i \mid d_j)$ denotes the probability of q_i being triggered by the document word d_j .when 2 words are same, the probability equals 1. If they are different and the word d_j belongs to the triggered vocabulary set of query,

the probability equals the according parameter in the T_q,otherwise the probability is 0。

(6) $\frac{tf(q_i)}{cs}$ is used for data smoothing; here $tf(q_i)$ denotes times of query word q_i appearing in document set and cs denotes the total length of documents which contains the word q_i.

6 Conclusion and Future Work

This paper mainly describes the methods and procedures we took in participating in NTCIR4 CLIR track. We focus our experiments on evaluating the effectiveness of the language model based IR method. At the same time, in this paper we also propose the trigger language model based IR system to relax the independent assumption of words within the classical language model IR method. The analysis of the language model based IR method compared with VSM method is also presented in the paper. Our future work will focus on relative experiments to testify the effectiveness of trigger language model based IR system.

Acknowledgments

This work is supported by Beijing New Star Plan of Technology & Science (NO.H020820790130) and the National Science Fund of China under contact 60203007.

Reference

[1] J.Ponte and W.B.Croft , A Language Modeling Approach to Information Retrieval. In Proceedings of the 1998 ACM SIGIR Conference on Research and Development in Information Retrieval. Pages 275-281,1998.

[2]D.H.Miller, T.Leek and R.Schwartz. A hidden Markov model information retrieval system. In Proceedings of the 1999 ACM SIGIR Conference on Research and Development in Information Retrieval. Pages 214-221,1999.

[3]A.Berger and J.Lafferty. Information retrieval as statistical translation. In Proceedings of the 1999 ACM SIGIR Conference on Research and Development in Information Retrieval. Pages 222-229,1999.

[4]T.Hofmann. Probabilistic latent semantic indexing. In Proceedings of the 1999 ACM SIGIR Conference on Research and Development in Information Retrieval. Pages 50-57,1999

[5]S.Deerwester,S.T.Dummais etc. Indexing by latent semantic analysis. Journal of the Society for Information Science,41(6):381-407,1990

[6]M. Srikanth and R. Srihari. Biterm Language Models for Document Retrieval. In Proceedings of the 2002 ACM SIGIR Conference on Research and Development in Information Retrieval. 2002

[7]R. Jin, A.G. Hauptmann and C. Zhai. Title Language Model for Information Retrieval. In Proceedings of the 2002 ACM SIGIR Conference on Research and Development in Information Retrieval. 2002

[8]C Zhai and J Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In Proceedings of the 2001 ACM SIGIR Conference on Research and Development in Information Retrieval. 2001