Investigation of the Lambda Parameter for Language Modeling Based Persian Retrieval

Hadi Amiri, Ashkan Zarnani, Mahbod.Tavallaee, Sadra Abedinzadeh, Masoud Rahgozar, Farhad Oroumchian

Database Research Group, School Of ECE, University Of Tehran, Tehran, Iran
Department of Information Technology, University of Wollongong, Dubai, UAE

Abstract

Language modeling is one of the most powerful methods in information retrieval. Many language modeling based retrieval systems have been developed and tested on English collections. Hence, the evaluation of language modeling on collections of other languages is an interesting research issue. In this study, four different language modeling methods proposed by Hiemstra [1] have been evaluated on a large Persian collection of a news archive. Furthermore, we study two different approaches that are proposed for tuning the Lambda parameter in the method. Experimental results show that the performance of language models on Persian text improves after Lambda Tuning. More specifically Witten Bell method provides the best results.

1. Introduction

The need for effective methods of automated information retrieval has increased because of the tremendous explosion in the amount of unstructured text data. For this purpose many approaches and methods have been developed [3], [16], [15], [10]. One of the most powerful and modern methods in information retrieval is language modeling. This method applies the technique of estimating the language model of each document in the collection. The major advantage of the language modeling approach is that it is non-parametric and integrates document indexing and document retrieval into a single model. In this approach, collection statistics such as term frequency, document length and document frequency are integral parts of the language model and are not used heuristically as in many other approaches. In addition, length normalization is implicit in the calculation of the probabilities and does not have to be done in an ad hoc manner.

The basic language modeling approach was initially proposed by Ponte et. al [2]. Hiemstra extended this basic approach by introducing the concept of importance of a query term [4]. The importance of a query term is an unknown parameter that explicitly models which of the query terms are generated from the relevant documents and which are not. Later another approach was proposed for the estimation of the language model parameters, called parsimonious language models [5]. Parsimonious language models need fewer (non-zero) parameters to describe the documents. Hiemstra in [1] proposed four methods of language modeling approach to information retrieval. The results showed that these methods have good performance on TREC collections and outperform some other ad-hoc methods [1], [2], [5], [6].

Language modeling has been applied with success to many languages such as English and Arabic [6]. In this research we have implemented and evaluated all of the four different methods of language modeling proposed by Hiemstra. These methods are evaluated on Persian text using a large size collection of an Iranian news archive. To further investigate the performance of language modeling on Persian text, two methods, Witten Bell method [19] and Dirichlet smoothing method [20] have been used to tune the Lambda parameter. The authors in [22] have shown these methods work well for tuning the Lambda parameter in language modeling based Arabic retrieval. Our experimental results show that tuning by Witten Bell method produces best results and increases the average precision at least 6% compared to the other method.

To the best of our knowledge only one work [7] is done on tuning Lambda for Persian language modeling. The major shortcomings of that work are the small size of the collection. In this work, we use a
standard and large size collection named Hamshahri Collection2 [8].

Our experimental results show that the retrieval precisions of all the four methods are comparable to each other. Furthermore, the results suggest that the Witten Bell method [19] is the best method to compute the value of the Lambda parameter.

In section 2 language modeling approach to information retrieval and the four Himstra’s models will be explained. Section 3 describes the collection that is used for experiments. The experimental results and comparisons are presented in section 5. Finally, the paper ends with the conclusions and future works provided in section 6.

2. Language Modeling Approach to Information Retrieval

Statistical language models have been around for quite a long time. They were first applied by Andrei Markov to model letter sequences in works of Russian literature [3].

In language modeling for each document in the collection the probability of generating the user request from that document should be defined. Documents are ranked according to this probability. Considerer \( P(D=d) \) as the prior probability of relevance of the document \( d \) which is the document that the user has in mind. For example \( P(D=d) \) could be estimated as:

\[
P(D = d) = \frac{\sum tf(t,d)}{\sum_i tf(t,k)}.
\]

Where \( tf(t,d) \) is the frequency of query term \( t \) in document \( d \) and the denominator is sum over all term frequencies in all documents.

The most obvious problem with this estimation is that it may assign a probability of zero to a document that is missing one or more of query terms [1], [2], [4], [5]. In addition, it is some what non-logical to have \( P(D=d)=0 \), i.e., the fact that a document does not contain a query term should not make that document non-relevant [2]. This problem is called sparse data problem. Hence, in information retrieval we need to assign some weight to a document in the collection even if a given query term dose not appear in the document. For this purpose Hiemstra considered a smoothing parameter lambda (\( \lambda_i \)) for each query term \( i \) [1], [4]. This parameter denotes the importance of query terms and has a value between zero and one. By assigning \( \lambda_i \) to seen terms (the query terms that are in the document) and 1-\( \lambda_i \) to unseen terms (the query terms that are not in the document), each document \( d \) will be ranked by calculating the following probability:

\[
P(D = d) \prod_{i=1}^{n} ((1-\lambda_i) \cdot P(T = t_i) + \lambda_i \cdot P(T = t_i | D = d))
\]

There are different ways to define the probabilities used in Equation 2 which will be reviewed in section 3

2.1. Previous work

To the best of our knowledge three groups have studied the use of language modeling based information retrieval for Persian language. Taghva and his colleagues [7] studied the application of language modeling techniques to Persian retrieval. They developed a language model engine named HLM4 (the fourth model of Hiemstra) for Persian language based on Hiemstra’s method. In their study, they determined the optimal value of \( \lambda \) to be 0.0485. They estimated \( \lambda \) by running 60 queries on 1647 documents several times while varying \( \lambda \). We believe, the major shortcoming with this work is the low number of documents in the used collection [23], [24]. Their experiment compares the average precision of language modeling approach with one of the standard vector space models, namely Lnc.btc. Their results show that language modeling approach improves the precision of retrieval by an average %11 against the Lnc.btc vector model.

Table 1 summaries the overall average of the eleven point precisions for their results [7]. SS indicates that the method uses stop word removal and stemming while NSS indicates that only stop word removal is used.

<table>
<thead>
<tr>
<th>Cosine_NSS</th>
<th>Cosine_SS</th>
<th>HLM4-NSS</th>
<th>HLM4-SS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.180</td>
<td>0.211</td>
<td>0.220</td>
<td>0.234</td>
</tr>
</tbody>
</table>

The other study on performance of language modeling on Persian text is done in Faculty of Engineering, University of Tehran [9], [18]. They
used hundreds of different combinations of different retrieval models including a few language modeling methods and their combinations to find the best configuration for a Persian retrieval engine. They used a collection known as Qavanin which consists of 170000 short documents extracted from 100 years of laws passed by the Iranian parliament. One drawback in this study was that only 14 queries were employed for the evaluation. Also the collection itself was not a good representative of Persian text because it only contains documents in the law domain. In their setup the language models performance was 10-15% below the vector space model.

In [21], the authors investigated the performance of Persian retrieval by merging the results of four different language modeling methods (proposed by Hiemstra) and two vector space models with Lnu.ltu and Lnc.btc weighting schemes. For the evaluations in [21] λ was set to the value proposed in [7]. Their experiments on Hamshahri suggest the usefulness of language modeling techniques for Persian retrieval.

For the above reasons, we used a large general purpose collection and a large number of queries in our experiments and evaluated the different models.

### 2.2. Hiemstra Method

Hiemstra proposed four ways to specify the probabilities and parameters in Equation 2. He emphasizes in [1] that each query term that is not in the stop list will be considered equally important if there is no previous relevance information available for a query, i.e. none of the relevant documents has been identified yet. Hence, in this case the model has only one unknown parameter as \( \lambda \) which will be equal for each position \( i \) in the query. Hence, the unknown parameter will simply be called \( \lambda \) in the following.

The equations 3 through 6 show Hiemstra’s models.

In the above four equations, \( tf(t,d) \) is the frequency of query term \( t \) in document \( d \) and \( cf(t) \) is collection frequency of query term \( t \). \( df(t,d) \) is the total number of terms in document \( d \) or length of document \( d \), and \( \sum df(t) \) is total number of terms in the collection or collection length. \( df(t) \) is document frequency of query term \( t \) and \( \sum df(t) \) is defined by sum of document frequency for all terms in the collection which has a constant value [1]. \( \sum \lambda_i tf(t,k) \) is the total length of the collection.

For \( P(T=ti) \), LM 3 like LM 1 uses collection frequency and LM 4 like LM 2 uses document frequency. The differences between the four methods can be summarized as follows: Document frequencies are used instead of collection frequencies in LM 2 and LM 4. Document length correction is also added to LM 3 and LM 4. Hiemstra determined in a series of experiments that the LM 4 was optimal for English text [1].

We have implemented all of these four models on Persian text. For evaluation we considered three different \( \lambda \) values. The first one is 0.0485, the value that Taghva and his colleagues determined as the optimal value of \( \lambda \). The second is computed using Witten Bell method [19]

\[
\lambda = \frac{\sum tf(t,d)}{\sum tf(t,d) + N_{doc}}.
\]

| Table 2: Eleven point recall-precision result of LM 1-4. |
\[ \lambda = 0.048 \]

<table>
<thead>
<tr>
<th>At Recall</th>
<th>( \lambda ) by Witten Bell method</th>
<th>( \lambda ) by Dirichlet smoothing method</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.37</td>
<td>0.41</td>
</tr>
<tr>
<td>0.1</td>
<td>0.31</td>
<td>0.35</td>
</tr>
<tr>
<td>0.2</td>
<td>0.28</td>
<td>0.33</td>
</tr>
<tr>
<td>0.3</td>
<td>0.25</td>
<td>0.30</td>
</tr>
<tr>
<td>0.4</td>
<td>0.22</td>
<td>0.29</td>
</tr>
<tr>
<td>0.5</td>
<td>0.19</td>
<td>0.27</td>
</tr>
<tr>
<td>0.6</td>
<td>0.15</td>
<td>0.22</td>
</tr>
<tr>
<td>0.7</td>
<td>0.11</td>
<td>0.19</td>
</tr>
<tr>
<td>0.8</td>
<td>0.07</td>
<td>0.15</td>
</tr>
<tr>
<td>0.9</td>
<td>0.02</td>
<td>0.07</td>
</tr>
<tr>
<td>1.0</td>
<td>0.01</td>
<td>0.02</td>
</tr>
<tr>
<td>Average</td>
<td>0.18</td>
<td>0.24</td>
</tr>
</tbody>
</table>

where \( N_{doc} \) is the number of unique terms in the document. Hence, using this formula the value of \( \lambda \) would be equal or more than 0.5.

The third method is Dirichlet smoothing method [20]. Equation 8 shows this method (\( k \) is constant value, equal to 800):

\[
\lambda = \frac{\sum tf(t,d)}{\sum tf(t,d) + k}.
\]

In next section we will compare the performance of these four models with each other and with two vector space models.

3. Experimental Results

In this research we have used a standard and large size collection named Hamshahri Collection [8]. Hamshahri is an Iranian newspaper that has been publishing for over twenty years in Iran [11]. The collection contains 345 Megabytes of Persian text and includes the news documents from June 1996 to January 2003. Hamshahri Collection contains more than 160,000 different documents with more than 417,000 unique words. This collection has 60 queries and relevance judgments for top 20 relevant documents for each query. Older versions of this collection were used in other Persian information retrieval experiments [8].

The standard TrecEval tool which is provided by NIST is used for evaluation [13]. We hope evaluating precision of different retrieval models with this big collection could yield more acceptable and reliable results.

3.1. Results of the Hiemstra Method

Precision of the four models at eleven point recalls is computed using TrecEval tool. The values are calculated for top 100 documents.

As it is shown in Table 2, tuning the Lambda parameter with Witten Bell method produces the best result and the Dirichlet smoothing method has the lowest performance. The best method for each tuning is bolded. Fig. 1 shows the recall precision graph for six models of the LM with different \( \lambda \) tuning methods, LM 1 to LM 4 with Witten Bell Lambda tuning method, LM 2 with \( \lambda = 0.048 \) and LM1 with Dirichlet Lambda tuning method.

![Figure 1: Precision- Recall Graph for language models LM1 to LM4 with different tuning](image)

It is clear from Fig 1 that LM1 with Witten Bell Lambda tuning method (LM1_Ndoc) has the best performance and outperforms other methods.

To have a better understanding of the behavior of these models we looked at two more diagrams namely; Document Cut Off and Average-R-precision diagrams. Document Cut Off diagram shows the precision after 5, to 100 documents have been retrieved. Fig. 2 shows the Document Cut Off
The X-axis represents the six document cut-offs and Y-axis shows the precision.

For further investigation, we considered LM2 and look at the effect of different values of \( \lambda \) (as a measure for determining query term importance) on this method. We selected LM2 because this method has acceptable performance with all the three different tunings. If we set \( \lambda \) to 0.048, LM2 prefers shorter documents than longer ones for each query term. According to Equation 4, this method gives less weight (\( \lambda \)) to the frequency of query terms while gives high weight (1-\( \lambda \)) to the document length in the denominator. However, considering Equation 7 and the values listed in Table 1, we understand that the Witten Bell method gives more weight (\( \lambda \)) to the frequency of query terms and lesser weight (1-\( \lambda \)) to the document length in the denominator.

<table>
<thead>
<tr>
<th>Avg. Document Length</th>
<th>Avg. No.of Unique Terms</th>
<th>Avg. ( \lambda ) by Witten Bell method</th>
<th>Avg. ( \lambda ) by Dirichlet smoothing method</th>
<th>Avg. ( \lambda = 0.048 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVG</td>
<td>442.62</td>
<td>203.63</td>
<td>0.66</td>
<td>0.31</td>
</tr>
</tbody>
</table>

This method increases the importance of term frequency by considering the number of unique terms in a document and normalizing the weight of the document length.

**4. Conclusions and Future Works**

In this paper, we reported implementation and evaluation of a retrieval engine for Persian text based on four different language models proposed by Hiemstra. The performance of these methods were evaluated and compared to each other using a large size collection of a news archive named Hamshahri. Two methods for tuning the Lambda parameter are evaluated in this study and compared with the previously proposed Lambda value. Experimental results reveal that, tuning LM1 by Witten Bell method, LM1_Ndoc, produces the best results and improves the precision compared to the previous models. It would be interesting to investigate if there are other values for tuning that could provide a better performance on the Persian collections in general and on the Hamshahri collection in particular. In future we would like to investigate other methods for tuning the Lambda parameter such as EM-algorithms.

**5. References**