Implement Topic Relevance Model For Query Expansion

Li Ren
Computer and Information Sciences
University of Delaware
Newark, DE 19713, USA
renli@udel.edu

Abstract

Query expansion is one of most important technique in information retrieval where provides a series of methods to reform the query applying to retrieval in order to extract result more precisely. In the past, there are many kind of trails on this technique in different perspective. In this project, I’ll try to implement query expansion with topic models, where using topic modeling technique to help us to do query expansion. In this project, I implement the Relevance Model (RM) approach by topic model LDA, and compare the retrieval result with baseline and naive query expansion result.

1 Introduction

In information retrieval, the queries are formed by search engine users. Studies shown that the average length of query provided by search engine users is only around 2-3 words. So it is very difficult to understand the user’s intention behind the query. For example, if the query is "apple store", it is not clear to indicate whether the user want to search stores that sell fruit or the electronic product. Therefore one common problem of queries provided by users is that they are short and vague. Thus, it is very difficult to estimate exactly what user need. One important technique to overcome this problem is Query Expansion that provides a family of methods to add additional terms to original queries in order to provide more information about the user needed.

There are many approaches using query expansion to solve the sparsity problem of query. Generally, all these methods can be classified into two major classes: global methods and local methods. Global methods are the approaches that reformulate query terms independent of the query and its retrieval result, while local methods use documents that are retrieved from the original result. According the comparison of these two class, the performance of local methods are better than the global method.[4]

The intuition of this project is trying to use topic model to help us overcoming the sparsity problem made by naive query expansion methods. Base on this intuition, we base on the work of Xing[12] where a topic relevance model was estimated based on Relevance Model (RM). The basic idea behind topic relevance model is assuming the topic behind the queries and the documents users needs are highly similar. Thus they build a model that assume the words are generated by query with specific topics. Thus they rank the topics by their probability to be generated given queries.

In this project, I implement one of most common local approach of query expansion - Pseudo-Relevance Feedback (PRF), which apply top k documents that are relevant and extract their terms to extend queries. One improvement of PRF is to apply a simple version of Topic Relevance Model[12] to select expansion terms. I’ll compare these two method with baseline and analysis their performance.

2 Related work

Pseudo-Relevance Feedback (PRF) assume top k documents to be relevant and the query is extracted
from the feedback documents. The basic idea behind PRF is that the top $k$ documents returned from original query retrieval reflect the initial attention user want to search. The idea of using probabilistic model on information retrieval was provided in 1979 by Croft and Harper.[2] They also mentioned a fundamental problem of query expansion: topic drift, which is drawn by the possibility that the result of adding terms has no correlation with the initial meaning of the query. This situation happens when one or some of first $k$ feedback documents have little relationship with the query.

On the research of query expansion, several approaches have been estimated to overcome topic shift and improve the robustness PRF. In 1998, Mitra develop a method to select a subset of feedback documents instead of choosing top $k$ documents.[7] Come to 2004, one idea came from Cronen-Townsend and Croft provided a method to dynamically decide the chance to apply PRF instead applying it for all queries.[3] Similar approach was also estimated by Cao on 2008.[1] Tao and Zhai implement a retrieval model that compute verify the importance parameters of each feedback document.[10] and in 2010 Lv and Zhai proposed a positional relevance model that weight the documents by their distance to query terms, where the document nearer to query is assigned a higher weight.[6]

3 Topic Relevance Model

Proposed by X. Yi in 2009[12], topic relevance model was estimated based on the RM[5] approach, where replace the multinomial distributions of words by topic distributions. They made complicated model which smoothing both topic relevance models and topic relevance models to improve the performance of query expansion. In this section, the implementation of their method is introduced.

3.1 Relevance Model

Majority research works on IR is based on probability ranking principle[9], which suggested to rank documents $D$ by the odds of their being observed in the relevant class $P(D | R)/P(D | N)$, where $R$ represents a class of documents that relevant to user’s query. The formal relevance model $R$ is an accurate model of relevance. The performance of probability ranking principle can be achieved by ranking documents by their posterior probability as they belong to the relevant model $R$. If we make common document assumption (document is represented as a sequence of words) and word independence assumption, the document ranking can be represented as following:

$$\frac{P(D | R)P(w | R)}{P(D | N)P(w | N)}$$

Generally the relevance models are estimated by sampling approach. Lavrenko estimated a approximating of relevance model as following:

$$P(w | R) \approx P(w | q_1...q_k) = \frac{P(w, q_1...q_n)}{P(q_1...q_n)}$$

Then they introduced two sampling approach on different assumption. One assume that all query terms are generated from one multinomial distribution $M$, sample the $P(w, q_1...q_n)$ by its joint probability given the distribution $M$. Another method assume each query term is generated from different multinomial distributions which sampled from a set of distributions.

3.2 LDA-RM

Based on relevance model, Xing and Allan tried to estimated their relevance model by replace multinomial distributions with topic distribution $t_i$. The generation process of topic relevance model can be shown as the following conditional probability:

$$P_{TM}(w | q) = \sum_{t_i} P_{TM}(w | t_i) \times P_{TM}(t_i | q)$$

The basic idea behind the LDA-RM is as following steps: First ranks each topic $t$ by its probability of generating the query $q$, then uses the words within high ranked $k$ topics to calculate a query-specific topic for query expansion. To investigate
whether topics discovered by typical topic models in PRF, they learn topic model from top-k retrieved documents and calculate a set of topic to calculate the query-specific model for query expansion.

They also combined the original relevance model with the topic relevance model by a parameter $\gamma$. Thus the model can be described as following:

$$P_{TM}(w \mid q) = \sum_{D_i \in C} \left( \gamma P_{TM}(w \mid D_i) + (1 - \gamma)P_{TM}(w \mid D_i,q) \right) \times P(D_i \mid q)$$

Where $\gamma$ they eventually tuned for different topic models.

The method to sample $P_{TM}(w \mid D_i,q)$ of this topic relevance model can be inferred as following: the model can be described as following:

$$P_{TM}(w \mid D_i,q) = \sum_{t_m} P(w \mid t_j) \times P(t_m \mid D_i,q)$$

$$P(t_m \mid D_i,q) = \frac{P(t_m,D_i,q)}{P(D_i,q)} = \frac{P(t_m,D_i)P(q \mid t_m)}{P(q \mid D_i)}$$

This can be described as when using the topic model to compute a relevance model, each topic portion $P(t_m \mid D_i)$ in document $D_i$ should be weighted by the probability of query $q$ generated by this document. Since they used LDA to implement the topic model, they call this approach $LDA-RM$.

4 The Project

In this project, I implemented a simple version $LDA-RM$ method with both the model and test system. The institution of this project is to explore how topic model could help on the query expansion technique in order to overcome the topic drift problem met by naive QE technique.

For the implementation, I apply Parallel Threaded - LDA [8][11] with 200 topic and 1500 iterations sampling. I didn’t try other topic model since the limit of time and computational ability. For corpus, I use classic Wall Street Journal 1978-1992 (173,252 documents) corpus which has been widely used in NLP experiment and TREC tasks. The queries are taken from TREC topic 51-100 (field title only).

To implement the program, I built the IR system with the help of Lucene API and built topic model method with the Mallet API. And I used hash table to store the trained data of topic model in order to optimize the speed of each iteration. Different from original $LDA - RM$, I calculate $P(w \mid t_j)$ from $P(w,t_j \mid q)$ by training topic from whole corpus instead of training from top $k$ documents in each iteration. One reason is for speed optimization, otherwise it may take several hours for a single iteration. Another reason is the intention to explore the effect after choosing a wider range of collection instead the feedback documents (this will effect the result, see explain in analysis section).

Before indexing and topic model training, preprocessing techniques are applied to corpus. First remove the stop words from the corpus, such as "a, the, of, for, etc.". Then applied Porter stemming algorithm, a small string processing language designed for creating stemming algorithms for use in IR to converge different type of words.

For the IR system, the retrieval model I choose is the Bayesian smoothing using Dirichlet priors [13], a study of smoothing methods for language models applied to Ad Hoc information retrieval and set the default parameter $\mu$ to 2500.

4.1 Experiments

For the test of this method, I choose corpus from WSJ and quires from TREC topics corresponding to WSJ. The following evaluation methods are tested:

1. Precision at different standard recall levels.
   The precision average at different standard recall levels are common to used in comparison the performance of different systems and as the input for plotting the recall-precision graph. A interpolated recall-precision average is calculated as following:

   $$\frac{\sum_{i=1}^{n} P_{\lambda}}{n}$$

   where $P_{\lambda}$ is the precision at different recall level $\lambda$

2. Average precision over all relevant documents (MAP). This measurement reflect the overall performance of all relevant documents. It is the average of the precision value obtained after each relevant document is retrieved.

3. Precision s at different document cutoff values This precision computed after a given number
Table 1: Retrieval Performance with TREC topics 51-100 on evaluation of Recall-Precision (RP), Precision with cut off, mean average precision and NDCG@10. The first raw is the baseline (BL) performance with no query expansion. The last row is the test result of original relevance model (RM). Others are tests with different parameters as table 2.

<table>
<thead>
<tr>
<th>no. documents</th>
<th>no. topics</th>
<th>no. terms</th>
<th>no. documents</th>
<th>no. topics</th>
<th>no. terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td>5</td>
<td>20</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>5</td>
<td>20</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>10</td>
<td>5</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>40</td>
<td>10</td>
<td></td>
<td>100</td>
<td>20</td>
</tr>
</tbody>
</table>

Table 2: Parameters setting of 5 TREC tests. The changed parameters are the number of top feedback documents that applied in RM, the number of topics and the number of expansion terms used in QE.

4. Result and Analysis

On the retrieval result on the corpus (WSJ) for different set of queries. We can see that although LDA – RM performance well on some evaluation (Precision at low cut off) and ranking evaluation (NDCG), the overall performance on Precision (MAP) does not as well as our expectation. From the table we can figure out that the variance of number of top k topic does not have significant effect on the performance unless k is small. The performance of this model is mainly effected by the number of documents that used to compute the relevance.

In my point of view, there are several reasons for the performance on this project. First, the topics I applied in the model is learned from whole collection, which cause the diversity of topics is so large that does not perform well when applying into a specific query. The author explain this problem as well by indicating that the whole corpus are too coarse-grained to be useful for query expansion. Secondly, the probability $P(q \mid t_m)$ could not guarantee the realistic chance the model generate the query for specific query, because the sparsity of query. That means when the length of query is too limit to sample topics explicitly. One simple example of Q52 for this reason can be shown as following:

**Query terms:** 'south' 'african' 'sanctions'

**Expansion terms:** 'mexico' 'black' 'govern' 'mexican' 'polit'

We see that "mexico" and "mexican" has high...
probability to be sample because they have high relation with query word “south” in some topic. Thus more technique should be add on semantic analyzing of query terms.

5 Conclusions

In this project, I implemented LDA – RM, a relevance model that applied LDA topic model to generate document words. I used the method by Xing to sample and compute the RM model and build a IR system to test the performance of the retrieval result on corpus WSJ. The model was simplified since the limitation of computational ability and time. The result of the test is not as well as my expectation and I briefly analyzed the reasons behind the performance.

For further work that should be done is to implement the technique or new model that could better perform on relevance model to find high relevance topic with quires. Another possible improvement is try to add weights on the expansion terms according to their relevance with topic model. Besides, more topic models can also be tried on this relevance model.

Applying NLP techniques to IR problems has been a popular research topic for a long time. However, not much evidences in the past provided that NLP techniques have significant help to solve IR problem since most of IR models treat documents as a collection of random words. Topic modeling are one of a few techniques that shown the help to IR problems in many perspectives. From this point of view, it is interesting and valuable to do more research works on implementation of retrieval model or query expansion technique.

5 Conclusions

In this project, I implemented LDA – RM, a relevance model that applied LDA topic model to generate document words. I used the method by Xing to sample and compute the RM model and build a IR system to test the performance of the retrieval result on corpus WSJ. The model was simplified since the limitation of computational ability and time. The result of the test is not as well as my expectation and I briefly analyzed the reasons behind the performance.

For further work that should be done is to implement the technique or new model that could better perform on relevance model to find high relevance topic with queries. Another possible improvement is to add weights on the expansion terms according to their relevance with topic model. Besides, more topic models can also be tried on this relevance model.

Applying NLP techniques to IR problems has been a popular research topic for a long time. However, not much evidences in the past provided that NLP techniques have significant help to solve IR problem since most of IR models treat documents as a collection of random words. Topic modeling are one of a few techniques that shown the help to IR problems in many perspectives. From this point of view, it is interesting and valuable to do more research works on implementation of retrieval model or query expansion technique.

References


