PERFORMANCE COMPARISON OF LANGUAGE MODELS FOR INFORMATION RETRIEVAL

Shuaixiang Dai, Qian Diao and Changle Zhou
Department of Computer Science and Technology, Xiamen University, 361005, Xiamen, Fujian,China; Intel China Research Center, 100087, Beijing, China

Abstract: Vector Space Model (VSM), Statistical Language Model (SLM) and Inference Network are three distinguished language models. Instead of evaluating their performance directly, we estimate the information strategies founded on them using the known measures: precision and recall. What's more, we proposed the Sort Order Rationality (SOR) to make further performance comparison among different language models. All models are tested on a standard testing collection. Three important conclusions are attained:

(1). The IR model combining the statistical language modeling and inference network approaches is better than that only founded on statistical language modeling approach. What's more, it is also better than that based on vector space modeling approach.

(2). The performance of IR model based on VSM is similar to that based on SLM.

(3). The *Dirichlet priors* method often is a better option to smooth a statistical language model.

In some respects, these conclusions provide some experimental bases for constructing an efficient information retrieval system.

Key words: language model, language modeling approach, information retrieval

1. INTRODUCTION

While the language modeling approaches, or more simple the language models are more and more wildly used in nature language process, in particular, some new approaches to text information retrieval based on language modeling methods have emerged, which are quite different from traditional probabilistic approaches, and are fundamentally different from vector space methods, one of the attractive aspects of the language modeling approach is the potential for estimating the document model or document-to-query translation model in different ways. This paper supplies the performance comparison of the popular language models by means of evaluating the information retrieval strategies based on them. It's maybe beneficial to the search engine builder.

2. LANGUAGE MODELS FOR IR

There are a lot of language modeling approaches have been proposed such as vector space model (VSM), statistical language model (SLM), inference network and latent semantic analysis (LSA) etc. So there also occur lots of information retrieval (IR) strategies based on them. For example, tf*idf[1] approach and cosine similarity model founded on VSM, Okapi[2] and KL-divergence[3, 4] belonging to SLM, and Indri model[5] using inference network with statistical language modeling probabilities. Retrieval strategies on the basis of language modeling assign a measure of similarity between a document and a query. Commonly, the strategies are based on the notion that the more often terms occur in both the document and the query, the more "relevant" the document is deemed to be to the query.

A retrieval strategy is an algorithm that takes a query q and a set of documents $d_1, d_2, \cdots, d_n$, and scores the Similarity Coefficient of them by some $SC(q,d_i)$ $(1 \le i \le n)$ function for each of the documents. In general, any approach to the retrieval problem is decomposed into three basic components: (1) query representation; (2) document representation; and (3) matching of query representation and document representation.

2.1 TF*IDF

The vector space model has been widely used in the traditional IR field and most web search engines. Both documents and queries are represented by vectors, which are sets of terms with associated weights. A vector similarity function, such as the inner product, can be used to compute the similarity between a document and a query.

We assume each document and each query are represented by a term frequency vector $\vec{d} = (x_1, x_2, \cdots, x_n)$ and $\vec{q} = (y_1, y_2, \cdots, y_n)$ respectively, where n is the total number of terms or the size of vocabulary and x_i, y_i are the frequencies (i.e., the counts) of term t_i in d and q respectively. Given a collection C, where N is the total number of documents in C and n_t is the number of documents with term t. All terms in a query or a document are weighted by the heuristic TF*IDF weighting formula:

$$w_{t,d} = tf(x)idf(t)$$

Where $tf(x) = c(x)$ and $idf(t) = \log\frac{N}{n}$.

So the weighted vectors for $\bar{d}$ and $\bar{q}$ are:

$$\bar{d} = (tf_d(x_1)idf(t_i), tf_d(x_1)idf(t_i), \cdots, tf_d(x_n)idf(t_n))$$

$$\bar{q} = (tf_q(y_1)idf(t_i), tf_q(y_1)idf(t_i), \cdots, tf_q(y_n)idf(t_n))$$

The score of document $\bar{d}$ against query $\bar{q}$ is given by

$$s(\bar{d},\bar{q}) = \sum\nolimits_{i=1}^{n} tf_d(x_i)tf_q(y_i)idf(t_i)^2$$

2.2 Cosine similarity model

Cosine similarity measures the cosine of the angle between two vectors. Thinking of a document or a query as a vector, similarity is the angle between two vectors. The bigger angle indicates less similarity. In this case long documents do not have an unfair advantage any more. The similarity depends on pattern of word use but not on document length; every document is most similar to itself. Cosine similarity between vectors for document $\vec{d_j}$ and query $\vec{q}$ is:

$$w(\vec{d_j},\vec{q}) = \frac{\vec{d_j} \bullet \vec{q}}{\left|\vec{d_j}\right| \cdot \left|\vec{q}\right|} = \frac{\sum_{i=1}^{n}(w_{i,j}w_{i,q})}{\sqrt{\sum_{i=1}^{n} w_{i,j}^2 \sum_{i=1}^{n} w_{i,q}^2}}$$

Where $w_{i,j}$ is the weight of term i in document j. The inner product is often used to normalize the term weight. If the $w_{i,j}$ is the unnormalized weight, the normalized weight $w'_{i,j}$ is given by

$$w'_{i,j} = \frac{w_{i,j}}{\sqrt{\sum_{j} w_{i,j}^2}}$$

2.3 Okapi

Okapi is the name given to a family of experimental retrieval systems [6, 7]. It is based on the Robertson-Sparck Jones probabilistic model of searching [8]. A detailed summary of the contributions to TREC-1~9 by the Okapi system is presented in [9, 10]. Unlike TF*IDF, the Okapi method not only considers the frequency of the query terms, but also the average length of the whole collection and the length of the document under evaluation. The similarity can also be described as the inner product of the query vector and the document vector. The Okapi weighting function is based on the Robertson-Sparck Jones weight[8]:

$$w^{(idf)} = \log \frac{(r+0.5)/(R-r+0.5)}{(n-r+0.5)/(N-n-R+0.5)}$$

where N is the number of documents in the collection, n is the number of documents containing the term, R is the number of documents relevant to a specific topic, and r is the number of relevant documents containing the term. This weight reduces to an inverse document frequency (IDF) weight without relevance information (when $R = r = 0$).

The Okapi TF formula is implemented as

$$w^{(tf)} = \frac{(k_1+1)tf}{(K+tf)}$$

$$K = k_1((1-b) + b\frac{dl}{avdl})$$

The TF formula with query modification as relevance feedback is

$$w^{(tf)} = \frac{(k_1+1)tf}{(K+tf)} \cdot \frac{(k_3+1)qtf}{(k_3+qtf)}$$

where tf is the frequency of occurrence of the term within a specific document; qtf is the frequency of occurrence of the term within a specific query; dl and $avdl$ are respectively the document length and the average document length measured in arbitrary units, such as word or a sequence of words; and k_i, b are the constants used in BM functions(Best-match weighting function).

The Okapi weighting function (BM25) is

$$w_{t,d} = w^{(tf)} w^{(idf)}$$

which describes the contribution of term t to the relevance of document d.

2.4 KL-divergence retrieval model

The KL-divergence retrieval model essentially scores a document by computing the KL-divergence between the query language model and the document language model, in fact, which is the relative entropy of the query model with respect to the document model. The KL-divergence retrieval model was introduced in [11] as a special case of the more general risk minimization retrieval framework. Interestingly, it is similar to the vector space model, except that it uses probabilistic models, rather than ordinary term vectors to represent a document or a query.

Given two probability mass functions $p(x)$ and $q(x)$, the Kullback-Leibler divergence (or relative entropy) between p and q , denoted $D(p \| q)$, is defined as

$$D(p \| q) = \sum_x p(x) \log \frac{p(x)}{q(x)}$$

The relevance value of a document with respect to a query can be measured by the following KL-divergence function:

$$D(\hat{\theta}_Q \| \hat{\theta}_D) = -\sum_{\omega} p(\omega | \hat{\theta}_Q) \log p(\omega | \hat{\theta}_D) + cons(q)$$

$$cons\ (q) = -\sum_{\omega} p(\omega | \hat{\theta}_Q) \log\ p(\omega | \hat{\theta}_Q)$$

where $\hat{\theta}_Q$ and $\hat{\theta}_D$ are the estimated query and document language models respectively; $cons(q)$ is the entropy of the query model, which is a document-independent constant, can be dropped.

To avoid simple counting zero probabilities, we can introduce some smoothing methods to smooth the language model. Assume the general form of a smoothed model to be the following:

$$p(\omega | d) = \begin{cases} p_s(\omega | d) & \text{if word } \omega \text{ is seen} \\ \alpha_d p(\omega | C) & \text{otherwise} \end{cases}$$

where $p_s(\omega | d)$ is the smoothed probability of a word seen in the document is, $p(\omega | C)$ is the collection language model, and α_d is a coefficient controlling the probability mass assigned to unseen words, so that all probabilities sum to one. The representative smoothing methods include *the Jelinek-Mercer method*, *Bayesian smoothing using Dirichlet priors* and *absolute discounting*, which are detailed in [12].

2.5 Indri model

Unlike above statistical language models, Indri model combines the statistical language modeling[14, 15] and inference network[16] approaches to information retrieval. The resulting model allows structure queries similar to those used in INQUERY to be evaluated using language modeling estimates within the network, rather than *tf*idf* estimates. Figure 1 shows a graphical model representation of the network. As in the original inference network framework, documents are ranked according to $P(I | D,\alpha,\beta)$, the belief the information need I is met given document D and hyperparameters α and β as evidence.

Typically, in the statistical language modeling framework, a document is represented as a sequence of terms (tokens). Based on this sequence, a multinomial language model over the vocabulary is estimated. However, In Indri model, documents are represented as multisets of binary feature vectors. Each document is estimated by a multiple-Bernoulli model[17]. The multiple-Bernoulli model imposes the assumption that the features (r_i's) are independent, which of course may be a poor assumption depending on the feature set.

Indri model takes a Bayesian approach and imposes a multiple-Beta prior over the model (θ). The Beta is chosen for simplicity, as it is the conjugate prior to the Bernoulli distribution. Thus, $P(D|\theta) \sim MultiBernoulli(\theta)$ and $P(\theta|\alpha,\beta) \sim MultiBeta(\alpha,\beta)$. The belief at node θ is given by

$$P(\theta_i | D,\alpha,\beta) = \frac{P(D|\theta_i)P(\theta_i|\alpha_i,\beta_i)}{\int_{\theta_i} P(D|\theta_i)P(\theta_i|\alpha_i,\beta_i)} = Beta(\#(r_i,D)+\alpha_i,|D|-\#(r_i,D)+\beta_i)$$

for each i, $\#(r_i,D)$ is the number of times, feature r_i is set to 1 in document D 's multiset of feature vectors.

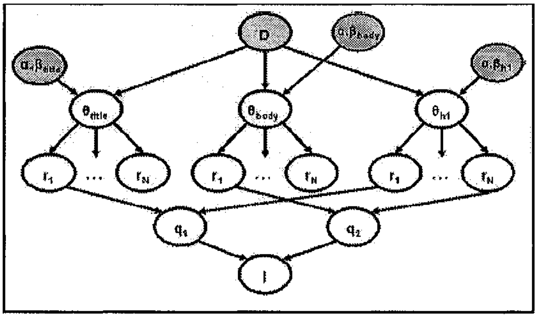

Figure 1. Indri's inference network retrieval model

3. EVALUATIONS AND COMPARISONS

To estimate a statistical language modeling approach for IR, In 1998 Ponte and Croft proposed using a smoothed version of the document unigram model to assign a score to a query, which can be thought of as the probability that the query was generated from the document model. This simple approach was remarkably effective. As developed further in [13]. In our experiments, we use three smoothed versions of KL-divergence and evaluated them respectively. Some known measures are taken, as well as a new but effective evaluation criterion.

3.1 Experimental Setup

We develop the evaluating programs based on Lemur Toolkit [18] and carry out a series of experiments. The English testing collection is CACM database (3204 documents, 64 queries and their answer set) that comes from Lemur downloading package. Sixty four queries with the right answers are also included and put to evaluate above language models. The following performance measures are considered in our evaluation:

- Non-interpolated average precision
- Interpolated precision at different recall levels
- Sort order by score of relevant document

Because some models are very similar in principle, their retrieval results are also very alike in precision. To distinguish the performance of these models clearly, we introduce the sort order by score of relevant documents to evaluate their retrieval performance at a different angle. Assuming a query q and a set of documents D, the Similarity Coefficient of q and d_i $(d_i \in D, 1 \le i \le n)$ is represented by $SC(q,d_i)$, we normalize the similarity coefficient by

$$s(q,d_i) = \frac{SC(q,d_i) - Min_{sc}}{Max_{sc} - Min_{sc}}$$

where Max_{sc} is the maximum in all $SC(q,d_i)$ $(1 \le i \le n)$, Min_{sc} is the minimum. The Sort Order Rationality (SOR) is defined as

$$SOR(q,D) = \frac{\sum_{j=1}^{R} s_j}{M}$$

For each j, s_j is the normalized similarity coefficient between q and one of its relevant documents $d_j (1 \le j \le n)$. R is count of relevant documents in retrieval result set. M is the count of documents relevant to query q in document collection.

The Sort Order Rationality (SOR) indicates the rational degree of sort order of relevant documents in retrieval result set. It also reflects the performance of IR model to some extent. If the relevant documents line in the front of all retrieval result documents, i.e. only retrieval less documents we can attain all relevant documents. If so, we think this kind of result is better. Thus the bigger the SOR value is, the more rational the result is, and the better performance the model shows.

For getting the true performance of above models, we use the simplest form of each model; don't consider feedback and other improvements over parameters. Especially, to Okapi, we set k_1=1.2, b=0.75 and k_3=7, which are suggested by Robertson in [19].

When using Indri to retrieval information, we also take *Dirichlet priors* to smooth the probabilistic model.

3.2 Results and Discussions

To every query, we compute all precisions and recalls when returning different number of result documents, for example from 1 to 1000. We attain the non-interpolated average precisions of all models, as follows.

Table 1, Comparison of average precisions

Models	Indri	KL-Dir	KL-JM	Okapi	TF*IDF	Cosine	KL-Abs
Average Precision	0.3481	0.3354	0.3207	0.3095	0.3057	0.2532	0.2459

From all recall-precision pairs computed above, we calculate 11 interpolated precisions when recall equals 0, 0.1, 0.2,···, 1 respectively, and attain the following diagram.

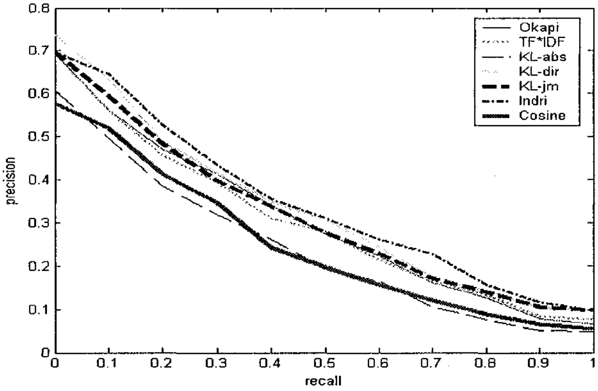

Figure 3. Interpolated recall-precision curves

From table 1 and figure 3, we can see that Indri model displays the best performance, but the Cosine and KL-Abs are contrary, they have the worse performance. The precisions of the TF*IDF, Okapi, KL-divergence smoothed by *Dirichlet priors* and KL-divergence smoothed by *Jelinek-Mercer method* are close, so are their performances.

In fact, whether TF*IDF, Okapi or KL-divergence smoothed by *Dirichlet priors* or *Jelinek-Mercer* is based on how often the term appears or does not appear in relevant documents and non-relevant documents. The betterment of weighting function can't bring essential improvement of performance, little if anything. This is why these models all return the similar precisions.

Indri not only use inference network but also add statistical language modeling probabilities, as compare with other models, it gives more information when retrieval. This may be one of the reasons why the performance of Indri model is best in our experiments.

To KL-divergence, different smoothing method brings different performance. It's obvious that *Dirichlet priors is* the best option to smooth a model.

Since the TF*IDF, Okapi, KL-divergence smoothed by *Dirichlet priors* and KL-divergence smoothed by *Jelinek-Mercer method* display the similar performance, we use average SOR to make further comparison. The average SOR of these models are given by Table 2.

Table 2, Average sort order rationalities when returning 1000 result documents

Models	KL-Abs	Okapi	KL-Dir	Indri	KL-JM	TF*IDF	Cosine
Average SOR	0.4309	0.4032	0.3977	0.3847	0.3582	0.3159	0.2819

From the table 2, we can see the average SOR values of Okapi and KL-Dir are close; in addition their precisions are also close, so we can consider their performances are alike. But their performances are better than those of KL-JM and TF*IDF, because SOR values of the former are bigger than those of the latter.

4. CONCLUSION

Lots of Language models are proposed and used in natural language process, especially in information retrieval. What's more, previous works showed all the models are effective respectively. However, these works did not tell us which one of them was better than other. In this paper, we selected five representative language models and designed a serious of experiments to compare the performance between them. The following points are concluded after the experiments:

The Indri model which combines the statistical language modeling and inference network approaches is better than that only founded on statistical language modeling (SLM) approach. What's more, it is also better than that based on vector space modeling (VSM) approach.

The performance of IR model based VSM is similar to that based on SLM.

The *Dirichlet priors* method often is a better option to smooth a statistical language model.

In some respects these conclusions are important to construct a high-powered information retrieval system; they also provide experimental bases for designing an efficient search engine.

ACKNOWLEDGEMENTS

Most of this work was done while the author was visiting Intel China Research Center; we thank Liu Juan and Zhang Yiming who is leader of

Scalable Statistical Computing Group at Intel China Research Center for discussing some issues about this paper.

REFERENCES

[1] Gerard Salton and M. J. McGill (1983): *Introduction to Modern Information Retrieval*, New York: McGraw-Hill.

[2] Robertson, S. et al. Okapi at TREC-3. Proceedings of the 3rd Text Retrieval Conference, 1994, pp 109-126.

[3] Zhai, C. and Lafferty, J. *Model-based feedback in the KL-divergence retrieval model.* In Tenth International Conference on Information and Knowledge Management (CIKM 2001), pages 403–410.

[4] C. Zhai and J. Lafferty, Model-based feedback in the language modeling approach to information retrieval, Tenth International Conference on Information and Knowledge Management (CIKM 2001), 2001.

[5] Metzler, D., Strohman T., Turtle H., and Croft, W.B., "Indri at TREC 2004: Terabyte Track" in the Online Proceedings of 2004 Text Retrieval Conference (TREC 2004).

[6] Special issue of Journal of Documentation 53 (1), 1997.

[7] Okapi projects home page. Available at http://web.soi.city.ac.uk/research/cisr/okapi/okapi.html.

[8] Robertson, S.E. and Sparck Jones, K. Relevance weighting of search terms. *Journal of the American Society or Information Science 27,* 1976, 129-146.

[9] Robertson, S.E. and Walker, S. (1999). Okapi/Keenbow at TREC-8. In TREC-9.

[10] Robertson, S.E. and Walker, S. (2000). Microsoft Cambridge at TREC-9: Filtering track. In TREC-9.

[11] J. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In 24th ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'01), 2001.

[12] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval, In *24th ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR'01), 2001.

[13] 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.

[14] W. B. Croft and J. Lafferty. *Language Modeling for Information Retrieval.* Kluwer, 2003.

[15] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *SIGIR 1998*, pp. 275–281.

[16] H. Turtle and W. B. Croft. Evaluation of an inference network based retrieval model. *TOIS*, 9(3):187–222, 1991.

[17] D. Metzler, V. Lavrenko, and W. B. Croft. Formal multiple Bernoulli models for language modeling. In *SIGIR 004*, pp. 540–541.

[18] http://www.lemurproject.org/.

[19] S.E. Robertson, and S.Walker (2000), “Okapi/Keenbow at TREC-8,” in E. Voorhees and D.K. Harman (editors), The Eighth Text REtrieval Conference (TREC-8), NIST Special Publication 500-246.