UDC 62-50

A. Lanevska

PROBABILISTIC RANKING IN THE LOCAL SEARCH

Introduction

Information retrieval (IR) has become an increasingly important research area due to the rapid growth of the social networks and the Internet. Therefore as soon as the large repositories of data have become available to a large public and due to development of social web the huge demand of effective local search emerged. According to the statistics gathered by the various search engines the average query length is approximately two words in 75% of cases. Obviously the precision rate is not very high and it doesn't often meet the requirements of the user.

Modern search engines do not handle all queries in appropriate way at the systems with frequent updates of information such as blogs, forums etc. In order to interact more naturally with humans, one has to deal with the potential ambivalence, impreciseness, or even vagueness of user requests, and has to recognize the difference between what a user might say or do and what she or he actually meant or intended.

One typical scenario of human-machine interaction in information retrieval is by natural language queries: the user formulates a request, e.g., by providing a number of keywords or some free-form text, and expects the system to show the relevant data in some amenable representation, e.g., in form of a ranked list of relevant documents. Many retrieval methods are based on simple word matching strategies to determine the rank of relevance of a document with respect to a query. Yet, it is well known that literal term matching has severe drawbacks, mainly due to the ambivalence of words and their unavoidable lack of precision as well as due to personal style and individual differences in word usage.

The study of information retrieval models has a long history. There is a great diversity of approaches and methodology developed, rather than a single unified retrieval model that has proven to be most effective; however, the field has progressed in two different ways: theoretical studies of an underlying model and empirical ones. The first direction is represented by various kinds of logic models and probabilistic models, the second one -by many variants of the vector space model. In some cases, there have been theoretically motivated models that also perform well empirically; for example, the BM25 retrieval function, motivated by the 2-Poisson probabilistic retrieval model, has proven to be quite effective in practice. Most of the effort in the field of text retrieval was put in the development of statistical retrieval models like the vector space model (proposed by Salton et al.), the classical probabilistic model (proposed by Robertson and Sparck Jones) and more recently the inference network model (proposed by Croft and Turtle) [1, 6]. The basic idea of language modeling [3] is to estimate a language model of each document and rank the documents by the likelihood of query according to the language model. This framework has its foundations in statistical theory and used in speech recognition and natural language processing. Another powerful approach is Latent Semantic Analysis (LSA) developed for automatic indexing and information retrieval that attempts to overcome existing problems by mapping documents as well as terms to a representation in the so called latent semantic space. Probabilistic latent semantic analysis [4] evolved from LSA via adding a sounder probabilistic model, is based on a mixture decomposition derived from latent class model. This results in a more principled approach which has a solid foundation in statistics.

The problem definition

The aim of this research is to develop a combined method of information ranking adapted for the given scope – social web such as search service for the blog considering advantages, drawbacks and constraints of existing methods and to describe the possible ways of methodology development further on.

The basic retrieval model

This paper defines a retrieval model based on combining a linguistically motivated methods of full text information retrieval and probabilistic latent semantic analysis in addition with query expansion and relevance feedback. The most important modeling assumption at the local search as well is that a document and a query are defined by an ordered sequence of words or terms.

The sample space

Language model assumes that a collection consists of finite number of textual documents [5]. The documents are written in a language that exists in a finite number of words or terms. Let *P* be a probability function on the joint sample space $\Omega_T \times \Omega_D$. Let Ω_D be a discrete sample space that contains a finite number of points *d* such that each *d* refers to an actual document in the document collection, *D* is a discrete random variable over Ω_D. Let Ω_T be a discrete sample space that contains a finite number of points terms *t* such that each t refers to an actual term that is used to represent the documents, *T* – a discrete random variable over Ω_T.

In other words, the random variable *D* refers to a document id and the random variable *T* refers to an index term.

Modeling documents and queries

Queries will be modeled as compound events. The single events that define the compound event are the query terms. In general the probability of a compound event does depend on the order of the single events [5]. A query of length n is modeled by an ordered sequence on *n* single terms *T_1, T_2, ..., T_n*. Given a document id *D* the probability of the ordered sequence will be defined by P (*T_1, T_2, ..., T_n/D*). Most practical models for information retrieval assume independence between index terms. Assuming conditional independence of terms given a document id leads to the following model.

$$P(T_1, T_2, ..., T_n / D) = \prod_i^n P(T_i / D) \qquad (1)$$

Note that the assumption of independence between query terms does not contradict the assumption that terms in queries have a particular order. The independence assumption merely states that every possible order of terms has the same probability.

The matching process

Equation (1) can be used directly to rank documents given a query *T_1, T_2, ..., T_n*. Let us rewrite (1) to a probability measure that explicitly ranks documents given a query:

P (D/T_1, T_2, ..., T_n). This measure can be related to (1) by applying Bayes' rule [3].

$$P(D / T_1, T_2, ..., T_n) = P(D) \frac{P(T_1, T_2, ..., T_n / D)}{P(T_1, T_2, ..., T_n)} = P(D) \frac{\prod_i^n P(T_i / D)}{P(T_1, T_2, ..., T_n)} \qquad (2)$$

It seems alluring to make the assumption that terms are also independent if they are not conditioned on a document D.

Since $\sum_d P(D = d / T_1, T_2, ..., T_n) = 1$ can scale the formula using a constant C such that

$$\frac{1}{C} = \sum_{d=1}^{D} P(D = d \cap T_1, T_2, ..., T_n)$$

we obtain

$$\sum_d P(D = d / T_1, T_2, ..., T_n) = CP(D) \prod_{i=1}^n P(T_i / D) \qquad (3)$$

Equation (3) defines the ranking formula of the linguistic motivated probabilistic retrieval model if we assume term independence.

Estimating probabilities from sparse data

Perhaps the most straightforward way to estimate probabilities from frequency information is maximum likelihood estimation. A maximum likelihood estimate makes the probability of observed events as high as possible and assigns zero probability to unseen events [3]. This makes the maximum likelihood estimate unsuitable for directly estimating P (*T/D*). One way of removing the zero probabilities is to mix the maximum likelihood model of P (*T/D*) with a model that suffers less from sparseness like the marginal P (*T*). It is possible to make a linear combination of both probability estimates so that the result is another probability function. This method is called linear interpolation:

$$P_{li}(T/D) = \alpha_1 P_{mle}(T) + \alpha_2 P_{mle}(T/D)$$
$$0 < \alpha_1, \alpha_2 < 1 \qquad (4)$$
$$\alpha_1 + \alpha_2 = 1$$

The weights α1 and α2 might be set by hand and $\alpha_1 P_{mle}(T)$ has to be smaller than $\alpha_2 P_{mle}(T/D)$ for each term *t*. This will give terms that do not appear in the document a much smaller probability than terms that do appear in the document.

Here let's define *N* as the number of documents in the collection, term frequency tf(*t,d*) as the number of times the term *t* appears in the document *d* and document frequency df(*t*) as the number of documents in which the term *t* appears. Given a specific document many terms will have a frequency of zero, so the term frequency suffers from sparseness. The document frequency of a term will never be zero, because terms that do not appear in any document will not be included in the model. The sparseness problem can be avoided by estimating P(*T/D*) as a linear combination of a probability model based on document frequency and a probability model based on term frequency as in (7):

$$P(T_i = t_i / D = d) = \alpha_1 \frac{df(t_i)}{\sum_t df(t)} + \alpha_2 \frac{tf(t_i,d)}{\sum_t tf(t,d)} \qquad (5)$$
$$P(D = d) = 1/n$$

Note that term frequency and document frequency are not derived from the same distribution. Although the term frequency can also be used to compute global information of a term by summing over all possible documents, this information will usually not be the same as the document frequency of a term, more formally:

$$df(t) \neq \sum_d tf(t,d) \qquad (6)$$

Equations (3) and (5) define the ranking algorithm.

Any monotonic transformation of the document ranking function will produce the same ranking of the documents. Instead of the product of weights we could therefore also rank the documents by the sum of logarithmic weights. The resulting vector product version of the ranking formula

$$\begin{aligned} similarity(Q,D) &= \sum_{k=1}^{l} w_{qk} \times w_{dk} \\ w_{qk} &= tf(t_k, q) \\ w_{dk} &= \log(1+\frac{tf(t_k,d)}{df(t_k)\sum_k tf(t,d)} * \frac{\alpha_2 \sum_t df(t)}{\alpha_1}) \end{aligned} \tag{7}$$

On first glance the constant seems to have little impact on the final ranking. But in fact, different values of α1 and α2 will lead to different document rankings. We will show some effects of different values of α1 and α2 on the ranking of documents, especially for short queries.

Probabilistic latent semantic analysis

Probabilistic latent semantic analysis (PLSA), also known as probabilistic latent semantic indexing (PLSI) is a statistical technique for the analysis of two-mode and co-occurrence data.[4] It has a solid statistical foundation since it is based on the likelihood principle. This implies in particular that standard techniques from statistics can be applied for questions like model fitting, model combination and complexity control. In addition, the factor representation obtained by PLSA allows to deal with polysemous words and to explicitly distinguish between different meanings and different types of word usage.

The core of PLSA is a statistical model which has been called aspect model. The latter is a latent variable model for general co-occurrence data which associates an unobserved class variable with each observation, i.e., with each occurrence of a word in a document. In terms of a generative model it can be defined in the following way:

- select a document *d* with probability P(*d*);
- pick a latent class *z* with probability P(*z/d*),
- generate a word *w* with probability P(*w/z*).

As a result one obtains an observed pair (*d, w*), while the latent class variable *z* is discarded. Translating this process into a joint probability model results in the expression

$$P(d,w) = P(d)P(w/d), \tag{8}$$

where $P(w/d) = \sum_{z\in Z} P(w/z)P(z/d)$ (9)

The aspect model is a statistical mixture model which is based on two independence assumptions: observation pairs (*d,w*) are assumed to be generated independently and the conditional independence assumption is made that conditioned on the latent class *z*, words *w* are generated independently of the specific document identity *d*. Given that the number of states is smaller than the number of documents (*K*<<*N*), *z* acts as a bottleneck variable in predicting *w* conditioned on *d*.

Notice that unlike the document clustering models document-specific word distributions P(*w/d*) are obtained by a convex combination of the aspects or factors P(*w/z*). Documents are not assigned to clusters, they are characterized by a specific mixture of factors with weights P(z/*d*). These mixing weights offer more modeling power and are conceptually very different from posterior probabilities in clustering models and (unsupervised) naive Bayes models.

Following the likelihood principle, one determines P(*d*), P(z/*d*), P(*w/z*) and by maximization of the log-likelihood function where denotes the term frequency. It is worth noticing that an equivalent symmetric version of the model can be obtained by inverting the conditional probability with the help of Bayes' rule[2]:

$$L = \sum_{d \in D} \sum_{w \in W} n(d,w) \log P(d,w) \tag{10}$$

The standard procedure for maximum likelihood estimation in latent variable models is the Expectation Maximization (EM) algorithm. EM alternates two steps:

- expectation (E) step where posterior probabilities are computed for the latent variables *z*, based on the current estimates of the parameters;
- maximization (M) step, where parameters are updated for given posterior probabilities computed in the previous E-step.

For the aspect model in the symmetric parametrization Bayes' rule yields the E-step

$$P(z/d,w) = \frac{P(z)P(d/z)P(w/z)}{\sum_{z'} P(z')P(d/z')P(w/z')} \tag{11}$$

which is the probability that a word *w* in a particular document or context *d* is explained by the factor corresponding to *z*. By standard calculations one arrives at the following M-step re-estimation equations

$$P(w/z) = \frac{\sum_{d} n(d,w)P(z/d,w)}{\sum_{d,w'} n(d,w')P(z/d,w')} \tag{12}$$

$$P(d/z) = \frac{\sum_{w} n(d,w)P(z/d,w)}{\sum_{d',w} n(d',w)P(z/d',w)} \tag{13}$$

$$P(z) = \frac{1}{R} \cdot \sum_{d,w} n(d,w)P(z/d,w) \tag{14}$$

$$R = \sum_{d,w} n(d,w) \tag{15}$$

Alternating expressions (11) with (12)-(15) defines a convergent procedure that approaches a local maximum of the log-likelihood in (10). So far we have focused on maximum likelihood estimation or, equivalently, word perplexity reduction.

Here we use a generalization of maximum likelihood for mixture models[7] – called tempered EM (TEM) – which is based on entropic regularization and is closely related to a method known as deterministic annealing. Essentially, one introduces a control parameter (inverse computational temperature) and modifies the E-step in (11) according to (16)

$$P_\beta(z/d,w)=\frac{P(z)[P(d/z)P(w/z)]^\beta}{\sum_{z'}P(z')[P(d/z')P(w/z')]^\beta}$$

It can be shown, that TEM minimizes an objective function known as the free energy and hence defines a convergent algorithm. While temperature-based generalizations of EM and related algorithms for optimization are often used as a homotopy or continuation method to avoid unfavourable local extrema, the main advantage of TEM in our context is to avoid over-fitting.

Query expansion and relevance feedback

The combination of different text representations and search strategies has become a standard technique for improving the effectiveness of information retrieval. We have described several representations of documents that could be used as evidence for *relevance* that denotes how well a retrieved set of documents (or a single document) meets the information need of the user. [1] Experiments with combinations of these representations show that, in general, using more than one representation improves retrieval effectiveness. They also show that when one source of evidence is weaker (less predictive of relevance) than the others, this must be reflected in the process of accumulating evidence or effectiveness will suffer.

Each additional piece of evidence that the query contains about the true information need can make a substantial difference to the retrieval effectiveness. This has long been recognized and is the basis of techniques such as relevance feedback [6], where user judgments of relevance from an initial ranked list are used to modify the initial query, and query expansion, which involves the automatic addition of new terms to the query.

In our case elaboration of the query is based on the thematic belonging of the documents, matched by the user as relevant ones at the first stage of search. Secondly query is expanded by the terms from the matched documents.

The query expansion is held in the following way [2]. Assume the user marked a set of the documents S. The weights of the words *w* are computed as

$$Q(w)=\sum_{d\in S,z\in Z}P(z)P(d/z)P(w/z) \qquad (17)$$

After that we sort all the words in descending order according to their weights and choose first several ones from list.

The final ranking scheme

The linguistically model has achieved very good retrieval results. Concerned mainly with performance numbers, recent work has shown the LM approach to be very effective in retrieval experiments, beating tf-idf and BM25 weights [6]. Nevertheless, there is perhaps still insufficient evidence that its performance so greatly exceeds that of a well-tuned traditional vector space retrieval system as to justify changing an existing implementation. The LM approach assumes that documents and expressions of information needs are objects of the same type, and assesses their match by importing the tools and methods of language modeling from speech and natural language processing. The resulting model is mathematically precise, conceptually simple, computationally tractable, and intuitively appealing. On the other hand, like all IR models, we can also raise objections to the model. The assumption of equivalence between document and information need representation is unrealistic. Usually LM approaches use very simple models of language. This model, however, does not solve the problem of retrieval the documents from the same topic.

As mentioned above PLSA allows to deal with polysemous words and to explicitly distinguish between different meanings and different types of word usage. However, it is reported that the aspect model used in the probabilistic latent semantic analysis has severe over-fitting problems. The number of parameters grows linearly with the number of documents. In addition, although PLSA is a generative model of the documents in the collection it is estimated on, it is not a generative model of new documents. Over-fitting occurs when a statistical model describes random error or noise instead of the underlying relationship. A model which has been over-fit will generally have poor predictive performance, as it can exaggerate minor fluctuations in the data.

In order to avoid over-fitting, it is necessary to use additional techniques, that can indicate when further training is not resulting in better generalization.

Therefore, in order to overcome the inconsistencies in these models, I propose the final ranking scheme combining all the approaches described above in the following way: we consider a linear combination of the linguistically motivated similarity score (7) (weight λ) and one derived from the latent space representation (weight $1-\lambda$) as suggested in (10) where $(0<\lambda<1)$.

$$(1-\lambda)P'(w/d)+\lambda\sum\nolimits_{z}P(w/z)P(z/d) \qquad (18)$$

Despite a synonym is not present at the document, nevertheless a conditional probability obtained by PLSI that it may appear can be a non-zero value.

Conclusions

Information retrieval (IR) has become an increasingly important area due to the rapid growth of the social networks and the Internet. However, search engines sometimes do not process a query properly in the systems with frequent updates of information such as blogs, forums etc.

This paper has presented a combining approach to the information retrieval at the local search via linear combination of the two powerful ranking frameworks: linguistically motivated modeling and probabilistic latent semantic analysis. The latter model incorporates also the relevance feedback and query expansion. These methods are the part of natural language processing techniques which slightly improve the performance of text retrieval. As a benefit there is dealing with polysemous words and to explicitly distinguish between different meanings and different types of word usage. Further investigation is needed to incorporate a collaborative filtering as a very popular tool at the local web services into the model.

References

1. *W. B. Croft*. Combining approaches to information retrieval. In Advances in Information Retrieval: Recent Research from the Center for Intelligent Information Retrieval // Kluwer Academic Publishers .-2000.- p.1-36.
2. *Бабий М. С., Чекалов А. П., Шевченко С. С.* Проектирование портала знаний по информационным технологиям // Вісник СумДУ. Серія Технічні науки.-2008.-№1.-С. 128-133.
3. Djoerd Hiemstra. A probabilistic justification for using tf.idf term weighting in information retrieval// International Journal on Digital Libraries, 3 (2).- 2000.- p.131-139.
4. *Hofmann.* Probabilistic latent semantic indexing// In Proceedings of the 22nd Annual Conference on Research and Development in Information Retrieval (ACM SIGIR).-1999.- p. 50-57.
5. *D. Hiemstra*. A linguistically motivated probabilistic model of information retrieval.// in Proc. the 2nd European Conference on Research and Advanced Technology for Digital Libraries, C. Nicolaou and C. Stephanidis, Eds.,Heraklion, Crete, Greece.- 1998.- p. 569–584.
6. *Manning C., Raghavan P., Schultze H.* Introduction to Information Retrieval.-Cambridge University Press.-2009.- p.544.
7. Mining the Web. Discovering knowledge from hypertext data. Soumen Chakrabarti. -2003.-Morgan Kaufmann Publishers.-p.345.