A Linguistically Motivated Probabilistic Model of Information Retrieval

Djoerd Hiemstra

University of Twente,
Centre for Telematics and Information Technology,
The Parlevink Language Engineering Group
P.O. Box 217, 7500 AE Enschede, The Netherlands
hiemstra@cs.utwente.nl
http://www.cs.utwente.nl/~hiemstra/

Abstract. This paper presents a new probabilistic model of information retrieval. The most important modeling assumption made is that documents and queries are defined by an ordered sequence of single terms. This assumption is not made in well known existing models of information retrieval, but is essential in the field of statistical natural language processing. Advances already made in statistical natural language processing will be used in this paper to formulate a probabilistic justification for using tf×idf term weighting. The paper shows that the new probabilistic interpretation of tf×idf term weighting might lead to better understanding of statistical ranking mechanisms, for example by explaining how they relate to coordination level ranking. A pilot experiment on the Cranfield test collection indicates that the presented model outperforms the vector space model with classical tf×idf and cosine length normalisation.

Keywords: Information Retrieval Theory, Statistical Information Retrieval, Statistical Natural Language Processing

1 Introduction

There are three basic processes an information retrieval system has to support: the representation of documents, the representation of a user's information need and the comparison of these two representations. In *text* retrieval the documents and the information need are expressed in natural language. Although text retrieval got by far the most attention in the information retrieval community, so far the success of natural language processing techniques was limited. 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. [11]), the classical probabilistic model (proposed by Robertson and Sparck Jones [7]) and more recently the inference network model (proposed by Croft and Turtle [3]).

The application of natural language processing techniques in combination with these models has solid but limited impact on the performance of text retrieval[1] [13]. The research does however provide little insight to the question *how* to use natural language processing. Natural language processing modules are usually considered as preprocessing steps, that is, they are not included in the model itself. This paper attempts to formulate a model that captures statistical information retrieval and statistical natural language processing into one unifying framework. It is the model itself that explicitly defines how documents and queries should be analysed. This seems a rather trivial requirement, but we claim that this is not the general idea behind the existing models for information retrieval. The (implicit) assumption made by these retrieval models is that some procedure, either manual or automatic, is used to assign index terms to documents. It is the result of this procedure that can be reflected by the model, not the procedure itself.

This paper is organised as follows. In section 2 the basic linguistically motivated retrieval model will be presented. Section 3 will give a new probabilistic interpretation of tf×idf term weighting by using estimation procedures developed in the field of statistical natural language processing. Finally, section 5 will present conclusions and plans for future work. These plans include the development of a model for phrases and the development of a model for cross-language information retrieval.

2 The Basic Retrieval Model

This paper defines a linguistically motivated model of full text information retrieval. The most important modeling assumption we make is that a document and a query are defined by an ordered sequence of words or terms.[2] This assumption is usually not made in information retrieval. In the models mentioned in the introduction documents and queries are modeled as unordered collections of terms or concepts. In the field of statistical natural language processing the word order assumption is essential for many applications, for instance part-of-speech tagging, speech recognition and parsing. By making the 'ordered sequence of terms assumption'

[1] Retrieval performance is usually measured in terms of precision (the fraction of the retrieved documents that is actually relevant) and recall (the fraction of the relevant documents that is actually retrieved).

[2] In the linguistically motivated model *terms* and *words* are equivalent, both expressions will be used in this paper. A classical index term that consists of more than one word will be called a *phrase*.

we will be able to use advances already made in statistical natural language processing. In this section we will define the framework that will be used in the subsequent sections to give a probabilistic interpretation of tf×idf term weighting.

2.1 The sample space

We assume that a collection consist of finite number of textual documents. The documents are written in a language that exist of a finite number of words or terms.

Definition 1 Let P be a probability function on the joint sample space $\Omega_D \times \Omega_T \times \Omega_N$. 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. Let D be discrete random variable over Ω_D. Let Ω_T be a discrete sample space that contains a finite number of points t such that each t refers to an actual term that is used to represent the documents. Let T be a discrete random variable over Ω_T. Let N be a discrete random variable over the sample space Ω_N. N will refer to the user's information need.

We will typically use the probability distribution $P(D|N)$ to rank documents given an information need. The random variable D will refer to an abstract representation of a document rather than to its physical representation which is modeled by index terms T.

We did not define the complete sample space Ω_N. The information need N is internal to the user and it is not known how it relates to other information needs. In practice we will usually only consider the conditional probability of terms and documents given one information need N, that is, in practice the model will be defined for one information need only.

2.2 Modeling documents and queries

A query and a document will be modeled as compound events. A compound event is an event that consists of two or more single events, as when a die is tossed twice or three cards are drawn one at a time from a deck [6]. In general the probability of a compound event does depend on the order of the events. For example a document of length n will be modeled by an ordered sequence of n random variables $T_1, T_2, \cdots, T_n$. The probability of the ordered sequence will be defined by $P(T_1, T_2, \cdots, T_n|D)$. Most practical models for information retrieval assume independence between index terms. This leads to the following model of documents.

$$P(T_1, T_2, \cdots, T_n|D) = \prod_{i=1}^{n} P(T_i|D) \tag{1}$$

Note that the assumption of independence between terms in documents does not contradict the assumption that terms in documents have a particular order. The independence assumption merely states that every possible order of terms has the same probability. It is made to illustrate that a simple version of the linguistically motivated model is very similar to existing information retrieval models. In the near future we hope to publish a version of the model in which the independence assumption does not hold: a model that uses phrases.

A query of length l will be modeled by an ordered sequence of l random variables $T_1, T_2, \cdots, T_l$. The probability of the ordered sequence will be defined by $P(T_1, T_2, \cdots, T_l|N)$. This probability can be viewed at as describing the process of a user formulating a query of length l, beginning with the word T_1 and ending with the word T_l. Unlike the modeling of documents, queries will not be modeled by assuming independence between query terms. Instead the query will get a probability of 1 if the user formulates only one query and some smaller probability if the user formulates more than one query. In a practical information retrieval system there should be some kind of interactive process between system and user in which one or more queries with corresponding probabilities are defined.

Analogous to a user formulating a query, equation 1 can be viewed at as the author of a document D writing a document of length n, beginning with the word T_1 and finally ending with the word T_n.

2.3 The matching process

The matching process is modeled by the joint probability measure $P(D, N, \mathcal{T})$ in which $\mathcal{T}$ will refer to the compound event $T_1, T_2, \cdots T_l$. The following conditional independence assumption is made:

$$P(D, N|\mathcal{T}) = P(N|\mathcal{T})P(D|\mathcal{T}) \quad , \mathcal{T} = (T_1, T_2, \cdots T_l) \tag{2}$$

By definition 1 terms are mutually exclusive and therefore two compound events are mutually exclusive if they differ in at least one of the single events. This means that it is possible to sum over all possible compound events $\mathcal{T}$ to determine the ranking of documents given an information need (see e.g. Wong and Yao [16]).

$$P(D|N) = \sum_{\mathcal{T}} P(\mathcal{T}|N)P(D|\mathcal{T}) \quad , \mathcal{T} = (T_1, T_2, \cdots T_l) \tag{3}$$

All compound events that were not formulated as a query by the user will have zero probability ($P(\mathcal{T}|N) = 0$), so in practice it is sufficient to use equation 3 for the formulated queries only. In order to use the independence assumption of equation 1 we should rewrite $P(D|\mathcal{T})$ in equation 3 by applying Bayes' rule.

$$P(D|T_1, T_2, \cdots, T_l) = P(D)\frac{P(T_1, T_2, \cdots, T_l|D)}{P(T_1, T_2, \cdots, T_l)} \tag{4}$$

$$= P(D)\frac{\prod_{i=1}^{l} P(T_i|D)}{P(T_1, T_2, \cdots, T_l)} \tag{5}$$

Equation 4 is the direct result of applying Bayes' rule. Filling in the independence assumption of equation 1 leads to equation 5. It seems tempting to make the assumption that terms are also independent if they are not conditioned on a document D. This will however lead to an inconsistency of the model (see e.g. Cooper's paper on modeling assumptions for the classical probabilistic retrieval model [2]). Since $\sum_d P(D = d|\mathcal{T}) = 1$ we can scale the formula using a constant C such that $\frac{1}{C} = \sum_d P(D = d, \mathcal{T})$.

$$P(D|T_1, T_2, \cdots, T_l) = C\ P(D) \prod_{i=1}^{l} P(T_i|D) \tag{6}$$

Equation 3 and 6 define the linguistic motivated statistical retrieval model if we assume independence between terms in documents.

3 Estimating the probabilities

The process of probability estimation defines how probabilities should be estimated from frequency of terms in the actual document collection. We will look at the estimating process by drawing a parallel to statistical natural language processing and corpus linguistics.

3.1 Viewing documents as language samples

The general idea is the following. Each document contains a small sample of natural language. For each document the retrieval system should build a little statistical language model $P(T|D)$ where T is a single event. Such a language model might indicate that the author of that document used a certain word 5 out of 1000 times; it might indicate that the author used a certain syntactic construction like a phrase 5 out of 1000 times; or ultimately indicate that the author used a certain logical semantic structure 5 out of 1000 times.

One of the main problems in statistical natural language processing and corpus linguistics is the problem of sparse data. If the sample that is used to estimate the parameters of a language model is small, then many possible language events never take place in the actual data. Suppose for example that an author wrote a document about *information retrieval* without using the words *keyword* and *crocodile*. The reason that the author did not mention the word *keyword* is probably different from the reason for not mentioning the word *crocodile*. If we were able to ask an expert in the field of *information retrieval* to estimate probabilities for the terms *keyword* and *crocodile* he/she might for example indicate that the chance that the term *keyword* occurred is one in a thousand terms and the chance that the term *crocodile* occurred is much lower: one in a million. If we however base the probabilities on the frequency of terms in the actual document then the probability estimates of low frequent and medium frequent terms will be unreliable. A full text information retrieval system based on these frequencies cannot make a difference between words that were not used 'by chance', like the word *keyword*, and words that were not used because they are 'not part of the vocabulary of the subject', like the word *crocodile*. Furthermore there is always a little chance that completely off the subject words occur like the word *crocodile* in this paper.

We believe that the sparse data problem is exactly the reason that it is hard for information retrieval systems to obtain high recall values without degrading values for precision. Many solutions to the sparse data problem were proposed in the field of statistical natural language processing (see e.g. [5] for an overview). We will use the combination of estimators by linear interpolation to estimate parameters of the probability measure $P(T|D)$.

3.2 Estimating probabilities from sparse data

Perhaps the most straightforward way to estimate probabilities from frequency information is maximum likelihood estimation [6]. A maximum likelihood estimate makes the probability of observed events as high as possible and assigns zero probability to unseen events. 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:

$$P_{li}(T|D) = \alpha_1 P_{mle}(T) + \alpha_2 P_{mle}(T|D), \quad 0{<}\alpha_1{,}\alpha_2{<}1 \text{ and } \alpha_1{+}\alpha_2{=}1 \quad (7)$$

The weights α_1 and α_2 might be set by hand, in which case we would choose them in such a way that $\alpha_1 P_{mle}(T = t)$ is smaller than $\alpha_2 P_{mle}(T = t|D)$ for each terms t. This will give terms that did not appear in the document a much smaller probability than terms that did appear in the document. In general one wants to find the combination of weights that works the best, for example by optimising them on a test collection consisting of documents, queries and corresponding relevance judgements.

Table 1. frequency information

N_q	the number of queries formulated for the information need at hand
N_d	the number of documents in the collection
$tf(t, d)$	*term frequency*: the number of times the term t appears in the document d.
$df(t)$	*document frequency*: the number of documents in which the term t appears.

Table 1 lists the frequencies that are used to estimate the probabilities of the model. Two frequencies are particularly important, the term frequency and the document frequency. The term frequency of a term is defined by the number of times a term appears in a document and can be viewed at as local or document specific information. 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 is defined by the number of documents in which a term appears and can be viewed at as global information. (Sometimes document frequency is referred to as collection frequency.) The document frequency of a term will never be zero, because by definition 1, 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 equation 10:

$$P(\mathcal{T}|N) = \frac{1}{N_q} \quad , \mathcal{T} = (T_1, T_2, \cdots, T_l) \tag{8}$$

$$P(D = d) = \frac{1}{N_d} \tag{9}$$

$$P(T_i = t_i|D = d) = \alpha_1 \frac{df(t_i)}{\sum_t df(t)} + \alpha_2 \frac{tf(t_i, d)}{\sum_t tf(t, d)} \tag{10}$$

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: $df(t) \neq \sum_d tf(t, d)$.

3.3 Relation to tf×idf

The use of term frequency and document frequency to rank documents was extensively studied, especially by Salton et al. for the vector space model [10]. They argued that terms appearing in documents should be weighted proportional to term frequency and inversely proportional to the document frequency. They called weighting schemes that follow this approach *tf*×*idf* (term frequency × inverse document frequency) weighting schemes and the application in a model of information retrieval the *term discrimination model*. The combination of tf×idf weights and document length normalisation gave the best retrieval results on several test collections, but they were not able to justify their approach by probability theory (which is not a prerequisite for using it in the vector space model anyway):

> *... The term discrimination model has been criticised because it does not exhibit well substantiated theoretical properties. This in contrast with the probabilistic model of information retrieval ...*

The lack of theoretical justification of tf×idf weights did not keep developers of the probabilistic model and the inference network model from using them. Robertson et al. [8] justified the use of term frequency in the probabilistic model by approximating a ranking formula that is based on the combination of the probabilistic model and the 2-poisson model. There is however a more plausible probabilistic justification of tf×idf weighting which can be justified by the linear interpolation estimator of equation 10. This can be shown by rewriting. Multiplying the ranking formula defined by equation 6, 9 and 10 with values that are the same for each document will not affect the final ranking, so we can multiply the ranking formula by $df(t)$ and α_2 as follows:

$$\begin{aligned} P(D=d|T_1=t_1,\cdots) &\propto \prod_{i=1}^{n}(\alpha_1 \frac{df(t_i)}{\sum_t df(t)} + \alpha_2 \frac{tf(t_i,d)}{\sum_t tf(t,d)}) \text{ [by eq. 6,9 and 10]} \\ &\propto \prod_{i=1}^{n}(\alpha_1 \frac{1}{\sum_t df(t)} + \alpha_2 \frac{tf(t_i,d)}{\sum_t tf(t,d)\cdot df(t_i)}) \quad [\times \textstyle\prod_{i=1}^{n} \frac{1}{df(t_i)}] \\ &\propto \prod_{i=1}^{n}(\frac{\alpha_1}{\alpha_2 \sum_t df(t)} + \frac{tf(t_i,d)}{df(t_i)} \cdot \frac{1}{\sum_t tf(t,d)}) \qquad [\times \textstyle\prod_{i=1}^{n} \frac{1}{\alpha_2}] \end{aligned}$$

The resulting formula can directly be interpreted as tf×idf weighting with document length normalisation, because:

$$\frac{\alpha_1}{\alpha_2 \sum_t df(t)} \text{ is a small constant for any document } d \text{ and term } t$$

$$\frac{tf(t_i,d)}{df(t_i)} \text{ is the tf×idf weight of the term } t_i \text{ in the document } d$$

$$\frac{1}{\sum_t tf(t,d)} \text{ is the inverse length of the document } d$$

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 [7]. The result would be an additive model, which is more in correspondence with the way in which the existing models mentioned in the introduction usually are presented.

On first glance the constant $\alpha_1/\alpha_2 \sum_t df(t)$ seems to have little impact on the final ranking. But in fact, different values of α_1 and α_2 will lead to different document rankings. In section 3.5 we will show some effects of different values of α_1 and α_2 on the ranking of documents, especially for short queries.

Document length normalisation which is made explicit by $1/\sum_t tf(t,d)$ is also defined by the estimator of equation 9. The a priori relevance of a document estimated as in equation 9 can be defined by any other estimator, e.g. by using approaches as described by Singhal et al. for the vector space model [12].

3.4 A new informal definition of tf×idf weighting

Equation 10 gives rise to a new informal definition of tf×idf weighting. Giving an informal definition after the formal definition seems a bit useless, but we believe that it will help to understand what exactly makes tf×idf weighting successful. The classical definition of tf×idf weighting can be formulated as follows:

Definition 2 The weight of a term that appears in a document should increase with the term frequency of the term in the document and decrease with the document frequency of the term. Terms that do not appear in a document should all get the same weights (zero weights).

An alternative definition is based on equation 10. If we assume that $\alpha_1\, df(t_i)/\sum_t df(t)$ is much smaller than $\alpha_2\, tf(t_i, d)/\sum_t tf(t, d)$ then it can be formulated as follows:

Definition 3 The weight of a term that appears in a document should increase with the term frequency of the term in the document. The weight of a term that does not appear in a document should increase with the document frequency of the term.

An example may clarify the implications of both definitions. Suppose the user formulates the query *information retrieval* and there is no document in the collection in which the terms *information* and *retrieval* both appear. Furthermore suppose that the term *information* is much more common, i.e. has a higher document frequency than the term *retrieval.* Now the system will rank documents containing k occurrences of the term *retrieval* above documents containing k occurrences of the term *information.* The classical explanation would be as follows:

Explanation 1 The query term *retrieval* matches better with documents containing *retrieval* than the query term *information* matches with documents containing *information*, because *retrieval* has a higher inverse document frequency than *information.*

The alternative explanation would be that:

Explanation 2 The query term *information* matches better with documents not containing *information* than the query term *retrieval* matches documents not containing *retrieval* because *information* has a higher document frequency than *retrieval.*

There is no a priori reason to prefer one explanation above the other. However, the idea of definition 3, that global information is only used to weight terms of which there is no local information, might lead to better understanding of probabilistic term weighting in text retrieval.

3.5 The problem of non-coordination level ranking

There is a well known problem with statistical information retrieval systems that use tf×idf weighting: sometimes documents containing n query terms are ranked higher than documents containing $n + 1$ query terms. We will call this problem the *problem of non-coordination level ranking* in which the *coordination level* refers to the number of distinct query terms contained in a document. A coordination level ranking procedure will always rank documents containing $n + 1$ query terms above documents containing n query terms even if the top documents have little evidence for the presence of $n + 1$ query terms and lower-ranked documents have a lot of evidence for the presence of n terms.

According to studies of user preferences and evaluations on test collections the problem of non-coordination level ranking becomes particularly apparent when short queries are used [9]. In a lot of practical situations short queries are the rule rather than the exception, especially in situations where there is no or little user training like with Web-based search engines. For some research groups, the importance of coordination level is the reason for developing ranking methods that are based on the lexical distance of search terms in documents instead of on document frequency of terms [1, 4]. However, as pointed out by experiments of Wilkinson et al. [15], some tf×idf measures (e.g. like the measure proposed by Robertson et al. [8]) are more like coordination level ranking than others (e.g. like the measure proposed by Salton et al. [10]). Wilkinson et al. showed that weighting measures that are more like coordination level ranking perform better on the TREC collection, especially if short queries are used.

Following the results of Wilkinson et al. it might be useful to investigate what exactly makes a weighting measure "like" coordination level ranking. The following example may provide some insight. Suppose the user enters a small query of only two terms a and b. As in the previous example a might be the term *information* and b might be the term *retrieval*. Furthermore suppose that the document d_1 contains a lot of evidence for term a and no evidence for the term b; and that document d_2 contains little evidence of both terms. It can be shown that document d_1 will have a lot of evidence for a and none for b if $tf(a, d_1)$ is high, $tf(b, d_1) = 0$ and the length of d_1 is short. Document d_2 contains little evidence of a and b if $tf(a, d_2) = tf(b, d_2) = 1$ and if d_2 is a long document. The length of a document will be defined by $l(d) = \sum_t tf(t, d)$. Now the following equation defines the requirement for coordination level ranking, that is, the similarity of document d_1 to the query $a\ b$ should be smaller than the similarity of document d_2 to the query. The left hand side of the equation contains the similarity of the query compared to document d_1 and the right hand side contains the similarity of the query to document d_2. The similarities are defined by the ranking formula introduced in section 3.3 with $c = \alpha_1/\alpha_2 \sum_t df(t)$.

$$\begin{aligned}
(c + \frac{tf(a,d_1)}{df(a)l(d_1)})(c+0) \ &< \ (c + \frac{1}{df(a)l(d_2)})(c + \frac{1}{df(b)l(d_2)}) \\
c^2 + \frac{c\ tf(a,d_1)}{df(a)l(d_1)} \ &< \ c^2 + \frac{c}{df(a)l(d_2)} + \frac{c}{df(b)l(d_2)} + \frac{1}{df(a)df(b)l(d_2)^2} \\
\frac{c\ tf(a,d_1)}{df(a)l(d_1)} - \frac{c}{df(a)l(d_2)} - \frac{c}{df(b)l(d_2)} \ &< \ \frac{1}{df(a)df(b)l(d_2)^2} \qquad (11) \\
&\vdots \\
c \ &< \ \frac{l(d_1)}{l(d_2)(tf(a,d_1)df(b)l(d_2) - df(b)l(d_1) - df(a)l(d_1))}
\end{aligned}$$

Equation 11 shows that we can rewrite the requirement for coordination level ranking as a requirement for the constant c. If c is small enough than the problem of non-coordination level ranking will never occur. By changing the value of c the ranking formula can be adapted to different applications. If we are developing a Web-based search engine we might choose a relatively small value for c, but if we are developing a search engine for evaluation in TREC [14] we might choose a higher value of c. For the Web-based search engine we might define a collection specific lower bound of c by keeping track of the collection extrema like maximum term frequency and document frequency (*maxtf* and *maxdf*) and maximum and minimum document lengths (*maxl* and *minl*). If we fill in these extrema and the definition $c = \alpha_1/\alpha_2 \sum_t df(t)$ in (11) then the lower bound will be defined as follows on the ratio between α_1 and α_2.

$$\frac{\alpha_1}{\alpha_2} < \frac{minl \cdot \sum_t df(t)}{maxl \cdot (maxtf \cdot maxdf \cdot maxl - maxdf \cdot minl - minl)} \qquad (12)$$

Equation 12 defines a ranking formula that always produces coordination level ranking for queries of two words. For longer queries the bound will be lower and for queries with unrestricted length only $\alpha_1 = 0$ will guarantee coordination level ranking.

3.6 A plausible explanation of non-coordination level ranking

The arguments in the previous section showed the following. The smaller the value of the constant c, the more the ranking formula will behave like coordination level ranking. It is good to note that most tf×idf measures defined for the existing models of information retrieval include constants for which the arguments introduced above also hold (for instance the "+0.5" in the Robertson/Sparck Jones formula [7, 8]). However, the classical definition of tf×idf weighting (definition 2) does not give a plausible explanation of why and when non-coordination level ranking does happen. Using the new definition 3 and the fact that c is defined by the ratio $\frac{\alpha_1}{\alpha_2}$ we can give the following explanation of non-coordination level ranking when tf×idf weights are used.

Explanation 3 Non-coordination level ranking occurs if query terms that do *not* appear in a document are weighted too high compared to query terms that *do* appear in a document.

According to definition 3 terms that do not appear in a document are weighted proportional to the document frequency. If we choose a relatively high value for the constant α_1 then query terms that do not appear in a document will be weighted too high, possibly causing non-coordination level ranking.

4 A Pilot Experiment

There remains an important question: How does the model perform in an experiment with a test collection? The following pilot experiment is relatively weak because we used a relatively outdated test collection and compared the new model with a relatively outdated vector space weighting scheme. The results are however promising. The next step will be evaluation in this years Text Retrieval Conference TREC-7 [14].

4.1 Experimental results

In the experiment we implemented a linguistically motivated probabilistic retrieval engine and a standard vector space engine. Both engines used the same tokenisation and stemming of the words in the documents. As a test collection we used the Cranfield collection which was also used extensively in early experiments with the vector space model [10]. Table 2 lists the non-interpolated average precision averaged over 225 queries of the Cranfield collection for different values of α_1 and α_2.

Table 2. experimental results on the Cranfield collection

weight		avg. precision
$\alpha_1 = 0.05$	$\alpha_2 = 0.95$	0.3904
$\alpha_1 = 0.1$	$\alpha_2 = 0.9$	0.4016
$\alpha_1 = 0.2$	$\alpha_2 = 0.8$	0.4141
$\alpha_1 = 0.4$	$\alpha_2 = 0.6$	0.4249
$\alpha_1 = 0.6$	$\alpha_2 = 0.4$	0.4297
$\alpha_1 = 0.8$	$\alpha_2 = 0.2$	0.4325
$\alpha_1 = 0.9$	$\alpha_2 = 0.1$	0.4311
$\alpha_1 = 0.95$	$\alpha_2 = 0.05$	0.4252

To evaluate how our weighting scheme performs relatively to other tf×idf weighting schemes with document length normalisation we implemented the vector space model with tfc.nfx weighting as proposed by Salton and Buckley [10]. The non-interpolated average precision averaged over 225 queries of this system was 0.4032 on the Cranfield collection.[3] The linguistically motivated system performs better for quite a wide range of different values of α_1 and α_2. Experiments with the TREC collection, have to determine if the difference in performance is in fact a property of the respective ranking strategies.

4.2 Coordination level ranking

The Cranfield collection is a small collection (1398 documents) with a relative large number of queries (255 queries). Cranfield has the following collection extrema: The smallest document is 18 words long, the longest 354 words. The maximum term frequency is 28 and the maximum document frequency 729. Following the arguments of the previous section it is possible to calculate a lower bound on the ratio between α_1 and α_2 that will define coordination level ranking given a query of length 2. This leads to a lower bound of 0.000525 on the ratio between α_1 and α_2 which corresponds roughly to $\alpha_1 = 0.0005$ and $\alpha_2 = 0.9995$. Although correct, the lower bound introduced by equation 12 is obviously not very useful for identifying proper values for α_1 and α_2. There are several reasons that might explain why the system performs optimally for much higher values of α_1:

1. Coordination level ranking does not lead to good average precision on the Cranfield collection.

[3] Salton and Buckley [10] report a 3-point interpolated average precision of 0.3841. Our version of their system reaches a 3-point interpolated average precision of 0.4204 which is probably due to the use of a stemmer.

2. The system does produce coordination level ranking, but the bound on the ratio between α_1 and α_2 is too low to be of any use.
3. The system does produce coordination level ranking, but the bound is not useful because the collection does not have very small queries (the average query length is about 9.5 words).

Additional experiments have to point out which reason or reasons actually explain the experimental results the best.

5 Conclusion and Future Plans

This paper presented the linguistically motivated probabilistic model of information retrieval. Using estimation by linear interpolation which is often used in the field of statistical natural language processing we were able to present a probabilistic interpretation of tf×idf term weighting. We showed that this new interpretation leads to better understanding of the behaviour of tf×idf ranking. In a pilot experiment we showed that a system based on the derived model performs better on the Cranfield collection than a system based on a standard vector space model using classical tf×idf weights and cosine document length normalisation.

This paper did not present the linguistically motivated model for information retrieval in its full strength. Although we claim that the most important modeling assumption of the model is that documents and queries are defined by an ordered sequence of terms, the assumption is not essential for the claims made in this paper. In future papers we will investigate two major information retrieval issues that require natural language processing techniques. The first issue is the use of phrases in information retrieval. The second issue is the problem of cross-language information retrieval.

Acknowledgements

I would like to thank the following people for their support. Franciska de Jong, Paul van der Vet and Wilbert Kallenberg for general advice; David Hawking of the Australian National University for his advice on coordination level ranking. Wessel Kraaij of The Netherlands Organisation for Applied Scientific Research (TNO) for discussions on information retrieval.

References

1. C.L.A. Clarke, G.V. Cormack, and E.A. Tudhope. Relevance ranking for one to three term queries. In Proceedings of RIAO'97, pages 388–400, 1997.
2. W.S. Cooper. Some inconsistencies and misidentified modeling assumptions in probabilistic information retrieval. ACM Transactions on Information Systems, 13:100–111, 1995.
3. W.B. Croft and H.R. Turtle. Text retrieval and inference. In P. Jacobs, editor, Text-based Intelligent Systems, pages 127–156. Lawrence Erlbaum, 1992.
4. D. Hawking and P. Thistlewaite. Relevance weighting using distance between term occurrences. Technical Report TR-CS-96-08, The Australian National University, August 1996. http://cs.anu.edu.au/techreports/.
5. C. Manning and H. Schütze, editors. Statistical NLP: Theory and Practice, draft. http://www.sultry.arts.su.edu.au/manning/courses/statnlp/, 1997.
6. A.M. Mood and F.A. Graybill, editors. Introduction to the Theory of Statistics, Second edition. McGraw-Hill, 1963.
7. S.E. Robertson and K. Sparck Jones. Relevance weighting of search terms. Journal of the American Society for Information Science, 27:129–146, 1976.
8. S.E. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In Proceedings of the SIGIR'94, pages 232–241, 1994.
9. D.E. Rose and C. Stevens. V-twin: A lightweight engine for interactive use. In Proceedings of the 5th Text Retrieval Conference TREC-5, pages 279–290. NIST Special Publications, 1997.
10. G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. Information Processing & Management, 24(5):513–523, 1988.
11. G. Salton and M.J. McGill, editors. Introduction to Modern Information Retrieval. McGraw-Hill, 1983.
12. A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In Proceedings of the SIGIR'96, pages 21–29, 1996.
13. T. Strzalkowski and K. Sparck Jones. Nlp track at trec-5. In Proceedings of the 5th Text Retrieval Conference TREC-5, pages 97–101. NIST Special Publications, 1997.
14. E.M. Voorhees and D.K. Harman. Overview of the 6th text retrieval conference. In Proceedings of the 6th Text Retrieval Conference TREC-6. NIST Special Publications, 1998.
15. R. Wilkinson, J. Zobel, and R. Sacks-Davis. Similarity measures for short queries. In Proceedings of the 4th Text Retrieval Conference TREC-4, pages 277–286. NIST Special Publications, 1996.
16. S.K.M. Wong and Y.Y. Yao. On modeling information retrieval with probabilistic inference. ACM Transactions on Information Systems, 13:38–68, 1995.