Term-Specific Smoothing for the Language Modeling Approach to Information Retrieval: The Importance of a Query Term

Djoerd Hiemstra
University of Twente,
Centre for Telematics and Information Technology
P.O. Box 217, 7500 AE Enschede, The Netherlands
hiemstra@cs.utwente.nl

ABSTRACT

This paper follows a formal approach to information retrieval based on statistical language models. By introducing some simple reformulations of the basic language modeling approach we introduce the notion of *importance* of a query term. The importance of a query term is an unknown parameter that explicitly models which of the query terms are generated from the relevant documents (the important terms), and which are not (the unimportant terms). The new language modeling approach is shown to explain a number of practical facts of today's information retrieval systems that are not very well explained by the current state of information retrieval theory, including stop words, mandatory terms, coordination level ranking and retrieval using phrases.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Theory

Keywords

Information Retrieval, Formal Models, Language Models, Search Strategies

1. INTRODUCTION

Information retrieval research has a long tradition of experimental work; most of today's information retrieval research is firmly based on the scientific method. Over 80 % of the papers presented at SIGIR 2001 [6] report some experimental results.

In their SIGIR paper on the Okapi weighting algorithms, Robertson and Walker [25] distinguish two forms of approaches to information retrieval research: Firstly, approaches based on formal models, where the model specifies an exact formula to be tried empirically; and secondly ad-hoc approaches, where formulae are empirically tried because they seem to be plausible. In this paper we follow a third form of approaches to information retrieval research, suggested by Bruza and Huibers [3]: Approaches based on formal models, where the model specifies an exact formula that is used to proof some simple mathematical properties of the model. Such properties do not have to be tried or verified empirically on some test collection, because their truth can be shown to hold by some simple mathematics.[1]

1.1 Information Retrieval in Practice

User surveys show that about 85 % of the users of the internet use web search engines to locate information [16]. Let's have a look at some practical situations in which millions of people find themselves on a daily basis, those of web search engines.

Web search engines like Google or AltaVista provide their users with a simple text field to enter some words. They respond by providing a ranked list of references to web pages that are hopefully relevant to the user. The way this ranking is produced can be modeled reasonably well with a language modeling approach to information retrieval, including for instance Google's page ranking approach using statistics on hyperlinks [14]. But the quality of the ranking is not the main issue here. The main issue is that successful retrieval engines give the user a sense of control over the system, i.e. they provide a mechanism that gives users the possibility to override the basic ranking mechanism. This will be illustrated below by two simple web search engine examples.[2]

If we for instance enter the query `IT magazines` in Google, then the system will only retrieve documents containing the word "magazines", stating that "IT" is a very common word

[1] Note that we do not argue in any way that we do not believe in experimental work. On the contrary, experimentation is of the utmost importance in information retrieval research!

[2] Similar mechanisms exist in commercial online services as provided by e.g. Dialog and LexisNexis.

SIGIR'02, August 11-15, 2002, Tampere, Finland.

that is not included in the search.[3] It makes perfect sense to ignore the word "IT" because it matches virtually every (English) document on the internet. In this case however, it is clearly wrong. But there is no harm done: The proper response of the user is now to search for `+IT magazines`. By using the '+' operator, the user indicates that "IT" is an important word for this search that should not be ignored, even if the search engine would have decided otherwise on the basis of the word's statistics.

As a second example, consider e.g. entering the query `compassioned extrovert computers` in AltaVista. None of the web pages indexed by the system matches all three words, but the system will retrieve lots of documents that match two words out of three. The top ranked documents however, are documents that match the two words "compassioned" and "extrovert", i.e., many documents about people that possess these qualities, but no computers. This makes perfect sense because the word "computers" is much more frequent than "compassioned" or "extrovert", and it is standard practice to rank documents containing words that occur very frequently in the collection below documents containing infrequent words (see e.g. [30]). From the user's point of view however, it is wrong if it is computers he/she is looking for. But again there is no harm done if we enter the query `compassioned extrovert +computers`. As in the example above, the '+' operator indicates that the word "computers" is an important word, that should not be ignored on the basis of its statistics. The system will respond by retrieving only documents that match the words "compassioned" and "computers", or documents that match "extrovert" and "computers".

1.2 The Limitations of Statistics in Document Ranking

What's the point to all this? The examples show that there are limitations to the performance that can be achieved by ranking on the basis of word statistics. Often, the importance of a word for the final document ranking is reflected by its frequency distribution in the collection. Sometimes however, the statistics and the rules-of-the-thumb are wrong; as in the first example where "IT" is erroneously ignored based on its frequency; and as in the second example where "computers" is erroneously considered as less important than the other two words based on the fact that it is much more frequent on the world wide web.

Note that the solution to such problems is not developing even better statistical ranking algorithms. Statistics work *most* of the time: That's the beauty and the curse of statistics! In cases where statistical ranking is clearly wrong, the user (or perhaps the system) should be able to override the system's default behaviour by explicitly stating which words are the important words in the query. As shown in the examples, this is standard practice in the popular commercial retrieval systems already, but it is not reflected in any mathematical model of information retrieval.

In the following sections, we extend recent advances on the use of language models for information retrieval. In Section 2 we extend the language modeling approach to information retrieval by introducing the concept of importance of a query term. Our objective is to develop mathematical models that provide the mechanisms described in the examples above. In Section 3 we show the implications of query term importance by concentrating on mathematical properties of the language modeling approach that hold in any empirical setting. In Section 4, we will further generalize query term importance to a situation in which there may be different combinations of multiple models for each term.

2. LANGUAGE MODELING AND TERM IMPORTANCE

The language modeling approach to information retrieval defines a simple unigram language model for each document in a collection [21]. For each document D the language model defines the probability $P(T_1, \cdots, T_n|D)$ of a sequence of n query terms $T_1, \cdots, T_n$ and the documents are ranked by that probability. Essential in this approach is *smoothing* of the probabilities. Smoothing is the task of reevaluating the probabilities, in order to assign some non-zero probability to query terms that do not occur in a document. As such, smoothed probability estimation is an alternative for maximum likelihood estimation. The standard language modeling approach to information retrieval uses a linear interpolation smoothing of the document model $P(T_i|D)$ with a general collection model $P(T_i|C)$ [1, 10, 15, 17, 19, 29]. Linear interpolation smoothing needs a parameter λ which is set empirically on some test collection, or alternatively estimated by the EM-algorithm on a test collection [17, 19].

$$P(T_1, \cdots, T_n|D) \;=\; \prod_{i=1}^{n} \big((1-\lambda)P(T_i|C) + \lambda P(T_i|D)\big) \quad (1)$$

There are many other approaches to smoothing language models (see e.g. [12]), some of which have been suggested for information retrieval as well, for instance smoothing using the geometric mean and backing-off by Ponte and Croft [21]; and Dirichlet smoothing and absolute discounting suggested in a recent study by Zhai and Lafferty [32]. Zhai and Lafferty evaluated a number of approaches to smoothing on several large and small TREC collections. They find that different situations call for different approaches to smoothing. Interestingly, they explain their empirical results by suggesting that there is more to smoothing than just the reevaluation of probabilities, which they call the *estimation* role of smoothing. A second role of smoothing, which they call the *query modeling* role, is to "explain" the common and non-informative words in a query. All smoothing approaches have this dual role, but linear interpolation smoothing defines a fixed smoothing parameter across – and independent of – all documents, which is necessary for query modeling.

The linear interpolation smoothing as defined by Equation 1 can be explicitly modeled as a two-state hidden Markov model [17]. So, instead of looking at smoothing as an alternative to maximum likelihood estimation of probabilities (the estimation role), we can look at smoothing as a model with some hidden events in which all probabilities are maximum likelihood estimates (the query modeling role). In this paper we will take the query modeling role somewhat further, by explicitly defining a hidden event for each query term. Such a *term-specific* smoothing approach can derived as follows.

[3] Note that most retrieval systems do not distinguish upper case from lower case, and confuse the acronym "IT" with the word "it".

First we assume independence between query terms:

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

Then we introduce for each query term T_i a binary random variable I_i denoting the *importance* of a query term. Thus, $I_i = 1$ if the query term is important and $I_i = 0$ if the query term is unimportant. The importance of a query term is a hidden variable, so we have to marginalize it by summing over its possible values: 0 and 1.

$$= \prod_{i=1}^{n} \Big(\sum_{i_i \in \{0,1\}} P(T_i, I_i = i_i|D) \Big)$$

Furthermore, let's assume that the importance of a query term does not depend on the document model. This is motivated by the fact that the importance of a query term will be used to encode the information (and only the information) about a term that cannot be derived from the document statistics or the collection statistics.

$$= \prod_{i=1}^{n} \Big(\sum_{i_i \in \{0,1\}} P(I_i = i_i) P(T_i|I_i = i_i, D) \Big)$$

If we write the full sum over the importance values we get:

$$= \prod_{i=1}^{n} \Big(P(I_i = 0) P(T_i|I_i = 0, D) + P(I_i = 1) P(T_i|I_i = 1, D) \Big)$$

How does this equation relate to Equation 1? We use λ_i to denote the unknown probability of term importance, so $\lambda_i = P(I_i = 1)$ and therefore $(1 - \lambda_i) = P(I_i = 0)$; We make the assumption that the probability of an important query term is determined by the document language model, so we use $P(T_i|D)$ as a short-hand of $P(T_i|I_i = 1, D)$; and we make the assumption that the probability of an unimportant term is determined by the collection model, using $P(T_i|C)$ as a short-hand of $P(T_i|I_i = 0, D)$. This gives us:

$$P(T_1, \cdots, T_n|D) \;=\; \prod_{i=1}^{n} \Big((1 - \lambda_i) P(T_i|C) + \lambda_i P(T_i|D) \Big) \quad (2)$$

Equation 2 differs from Equation 1 by the fact that we have a separate smoothing parameter λ_i for each query term i. But maybe the correct view is that we did not use any smoothing of probabilities at all: We just formulated a bit more complex model, and all of the model's probabilities are defined by maximum likelihood estimates. This is visualized in Figure 1.

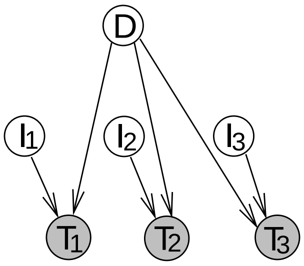

Figure 1: The language modeling approach as a Bayesian network

Figure 1 shows a graphical illustration of the model as a Bayesian network. In a Bayesian network, random variables are represented as nodes, and the arcs represent the model's conditional dependencies. Following the convention in [13], clear nodes represent unknown, hidden variables, and shaded nodes represent the observed variables; in this case the query terms.

3. MATHEMATICAL PROPERTIES OF TERM-SPECIFIC SMOOTHING

Term-specific smoothing provides the means to explain the examples described in Section 1. We will do so by first looking at the extreme values of the probability of term importance.

3.1 Stop words

The probability of term importance $\lambda_i = 0$ implies that we are absolutely certain that the query term is not important. It is easy to verify that any query term for which $\lambda_i = 0$, the document model $P(T_i|D)$ can be dropped from the calculation of the document score (because $\lambda_i P(T_i|D)$ will be zero whatever the value of $P(T_i|D)$). Therefore, we can simply ignore the query term all together, as is done with the word "IT" in the first example described in Section 1.

Words from the query that are ignored during the search are usually referred to as *stop words*. Words might be ignored because they occur very frequently in the collection. Frequent words might not contribute significantly to the final document score, but they do require a lot of processing power. The motivation for stopping is in this case based on a trade-off between speed and quality: the quality of the search results might be somewhat worse, but the processing speed will be much better [2, 4, 20].

Note however that words might be ignored for different reasons, that is, stop words are not always words that occur in many documents. Often, words are stopped if they occur in a list that enumerates words that usually carry little meaning, like for instance "the", "it" and "a". These words do also have a high frequency in English, but most publicly available stop lists contain some words which are not frequently used at all. For instance the stop list in [22], contains words like "hereupon" and "whereafter", which occur respectively two and four times in the TREC-8 ad hoc collection.

3.2 Mandatory terms

A query term for which the probability of term importance $\lambda_i = 1$ results in the following. For this query term, the collection model can be dropped from the calculation of the document score, because $(1-\lambda_i)P(T_i|C)$ will be zero whatever the value of $P(T_i|C)$. As a result, the probabilities of the document model are no longer smoothed with the probabilities of the collection model: Documents that do not match the query term are assigned zero probability, and since any multiplication with zero will be zero, the final document score will be zero as well no matter how well it matches the other query terms.

Assigning $\lambda_i = 1$ implies that we are absolutely certain that the relevant documents contain the term. A document will only get a non-zero probability if it matches the important term: Any document that does not contain the important term will not be retrieved. We will call a query term that should occur in every document retrieved a *mandatory* term. Usually, it is the user who decides which terms in his/her query are mandatory terms, for instance by using the '+' operator as described in the examples of Section 1.

3.3 Coordination level ranking

So, $\lambda_i = 1$ ensures that all retrieved documents contain the ith query term. This makes us suspect that for values close to 1, the ranking will be a *coordination level* ranking. Coordination level ranking is a partial ranking of the documents such that documents containing n query terms are always ranked above documents containing $n-1$ query terms.

According to studies of user preferences, users like systems that obey the conditions of coordination level ranking. Users have their own mental models of how search engines work or should work [18]. They may start to question the integrity of a search engine that ranks a document that matches one query term above documents that match two query terms; even if this seems reasonable from the query's semantics. For instance, it seems reasonable that for the query `big brown bear`, documents containing only "bear" are more likely to be relevant than documents about big brown 'whatevers'. However, users (read: customers) might consider the former result an error because it does not fit their mental model of how a search engine should work [27]. Obviously, users become particularly aware of such rankings if short queries are used. 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 search engines.

For systems that use some form of *tf.idf*-like term weighting, it is certainly not uncommon that documents containing k query terms are ranked below documents containing $k-1$ query terms. For some research groups, this is the main motivation for developing ranking methods that are based on other statistics, e.g. on the lexical distance of search terms in documents [5, 8]. As pointed out by experiments of Wilkinson et al. [31], some *tf.idf* measures behave more "like" coordination level ranking than others. For instance, the Okapi BM25 algorithm [25] behaves more like coordination level ranking than the Smart tfc.nfc algorithm [28]. They showed that weighting measures that are more like coordination level ranking perform better on TREC experiments, especially if short queries are used. Following their results, it might be useful to investigate what exactly makes a weighting algorithm behave like coordination level ranking.

A proof of coordination level ranking if λ approaches 1

As said, we suspect that high values of λ_i lead to coordination level ranking. Let us go back for a moment to the standard language modeling approach with one fixed λ for all query terms. This would be the system of our choice if no information on stop words, on mandatory terms, or on relevant documents (see Section 3.5) is available. First, let's have a look again at Equation 1:

$$P(T_1, \cdots, T_n|D) \;=\; \prod_{i=1}^{n}\big((1-\lambda)P(T_i|C) + \lambda P(T_i|D)\big)$$

Dividing the formula by $\prod_{i=1}^{n}((1-\lambda)P(T_i|C))$ will not affect the ranking because λ and $P(T_i|C)$ have the same value for each document.

$$P(T_1, \cdots, T_n|D) \;\propto\; \prod_{i=1}^{n}\big(1 + \frac{\lambda P(T_i|D)}{(1-\lambda)P(T_i|C)}\big)$$

Any monotonic transformation of the document ranking function will produce the same ranking of the documents. Instead of using the product of weights, the formula can as well rank documents by the sum of logarithmic weights.

$$P(T_1, \cdots, T_n|D) \;\propto\; \sum_{i=1}^{n}\log\big(1 + \frac{\lambda P(T_i|D)}{(1-\lambda)P(T_i|C)}\big)$$

The right-hand side of the equation defines a sum of log-likelihood ratios. Terms that do not match a document do not contribute to the sum [9, 32]. This gives us an easy way to formulate the requirement of coordination level ranking in terms of k matching terms, that is, a document that matches k terms should always have a higher score than a document that matches $k-1$ terms:

$$k\log\Big(1 + m\tfrac{\lambda}{1-\lambda}\Big) \;>\; (k-1)\log\Big(1 + n\tfrac{\lambda}{1-\lambda}\Big)$$

The left-hand side of the inequality is the matching score of a document that contains k query terms. The right-hand side of the inequality is the matching score of a document that contains $k-1$ query terms. In the inequality, m and n $(n, m > 0)$ replace the $P(T_i|D)/P(T_i|C)$ ratios of the matching terms. Note that the $P(T_i|D)/P(T_i|C)$ ratio is proportional to the frequency of occurrence of the term in a document and inversely proportional to the frequency of occurrence of the term in the collection, just as a *tf.idf* weight [9, 32]. For the simplicity of the proof, m is taken as the minumum of the $P(T_i|D)/P(T_i|C)$ ratios of the k matching terms of the document on the left-hand side, and n is taken as the maximum of the $P(T_i|D)/P(T_i|C)$ ratios of the $k-1$ matching terms of the document on the right-hand side. Proofing coordination level ranking for the extreme values of m and n will proof coordination level ranking for practical cases in which the ratio's values differ per matching term. Note that coordination level ranking might not be fulfilled if $n \gg m$.

It is easy to verify that the inequality holds if $k = 1$. For $k = 1$, the right-hand side is zero, and the inequality is true if $\lambda > 0$, no matter what the values of m and n are.

If $k > 1$, the k's might be moved to the left-hand side of the inequality and the rest to the right-hand side, resulting in:

$$\frac{k}{k-1} \;>\; \frac{\log\Big(1 + n\tfrac{\lambda}{1-\lambda}\Big)}{\log\Big(1 + m\tfrac{\lambda}{1-\lambda}\Big)}$$

In the following, we will proof that right-hand side of the inequality goes to 1 if λ approaches 1. So in the limiting case, the inequality will be true, because $k\,/\,(k-1) > 1$ for any bounded $k > 1$. So, we only have to show that for any fixed m and n:

$$\lim_{\lambda\to 1}\frac{\log\Big(1 + n\tfrac{\lambda}{1-\lambda}\Big)}{\log\Big(1 + m\tfrac{\lambda}{1-\lambda}\Big)} \;=\; 1$$

This can be shown as follows.

$$\frac{\log\left(1+n\frac{\lambda}{1-\lambda}\right)}{\log\left(1+m\frac{\lambda}{1-\lambda}\right)} = \frac{\log\left(\frac{1-\lambda+n\lambda}{1-\lambda}\right)}{\log\left(\frac{1-\lambda+m\lambda}{1-\lambda}\right)}$$

$$= \frac{\log(1-\lambda+n\lambda)-\log(1-\lambda)}{\log(1-\lambda+m\lambda)-\log(1-\lambda)}$$

Divding the numerator and the denominator by $-\log(1-\lambda)$ results in:

$$= \frac{1-\dfrac{\log(1-\lambda+n\lambda)}{\log(1-\lambda)}}{1-\dfrac{\log(1-\lambda+m\lambda)}{\log(1-\lambda)}}$$

which will in fact approach 1 if λ approaches 1, because $\lim_{\lambda\to 1}\log(1-\lambda+n\lambda)=\log n$, $\lim_{\lambda\to 1}\log(1-\lambda+m\lambda)=\log m$, and $\lim_{\lambda\to 1}\log(1-\lambda)=\infty$.

A value of λ close to 1 results in coordination level rankings. A high value implies that we are pretty confident that all query terms are important terms, which in general would be the case with short queries. For longer queries, it might no longer be justified to assume that all query terms are important, and a lower value of λ might be a better choice. In fact, this is supported by the experimental results of Zhai and Lafferty [32]: A high value of λ is a good choice for short queries, whereas a lower value is more appropriate for longer, verbose queries.

3.4 Discussion

The probability of term importance λ_i provides a mathematical model (or explanation) of a number of simple facts of information retrieval, if we consider its extreme values $\lambda_i = 0$, $\lambda_i = 1$ and $\lambda_i \to 1$ for all i:

- Sometimes systems ignore words, and the reasons to ignore them cannot always be explained by statistics.
- Often users/customers may want to restrict the retrieved list of documents to documents that match one or more specific terms, regardless of the frequency distributions of these terms.
- Users/customers may want to enforce a coordination level ranking of the documents, regardless of the frequency distribution of the terms.

We showed that it is possible to reason about these facts in terms of a language modeling approach to information retrieval. Note that we do not have to show empirically on a test collection that our language modeling system supports stop words, mandatory words and coordination level ranking: We are able to show by using some simple mathematics that this will be the case in any empirical setting.

3.5 Relevance feedback

As shown above, there are two extremes to explicitly modeling the importance of a term. At one end of the spectrum terms might be completely ignored if we assume that they are unimportant. At the other end of the spectrum terms are mandatory in the retrieved list of documents if we assume that they are important. However, in general we might expect the optimal values of λ_i to be somewhere between the extremes.

Finding optimal values for some unknown parameters in information retrieval is often addressed from the viewpoint of relevance feedback [24, 26]. Given some examples of relevant documents, what would be the probability of term importance for each term that maximizes retrieval performance? A standard approach to optimization in the language modeling field is the use of the Expectation Maximization (EM) algorithm [7]. The EM-algorithm will maximize the probability of the observed data given some training data. The EM-algorithm that maximizes the probability of the query given some relevant documents would be defined as follows.

$$\text{E-step:}\quad m_i = \sum_{j=1}^{r} \frac{\lambda_i^{(p)} \cdot P(T_i|D_j)}{(1-\lambda_i^{(p)})P(T_i|C) + \lambda_i^{(p)} P(T_i|D_j)}$$

$$\text{M-step:}\quad \lambda_i^{(p+1)} = \frac{m_i}{r}$$

The algorithm iteratively maximizes the probability of the query $T_1, \cdots, T_n$ given r relevant documents $D_1, \cdots, D_r$. Before the iteration process starts, the unknown parameters λ_i are initialized to their default values $\lambda_i^{(0)}$. Each iteration p estimates a new value $\lambda_i^{(p+1)}$ by first doing the E-step and then the M-step until the value of $\lambda_i^{(p+1)}$ is not significantly different from the value of $\lambda_i^{(p)}$ anymore.

Experimental results on a retrospective relevance weighting task show that this algorithm provides a relative performance gain that is as good as relevance feedback via the traditional probabilistic model [11]. In this study, the performance of the language modeling approach – with and without relevance feedback – is about 60 % higher than the performance of the probabilistic model.

4. A GENERALIZATION OF TERM IMPORTANCE

A simple but powerful generalization of the model presented so far is allowing the random variable I_i to have more than two realizations. Such an approach defines a linear combination of more than one document model and the background model; combining the standard document model with e.g. a model of the title words, or e.g. a model of anchor texts of hyperlinks pointing to the document.

Interesting from this perspective is the combination of the unigram document model with a bigram document model to explicitly include the search for phrases, as suggested by Miller et al. [17] and Song and Croft [29]. The model might be defined as a multi-state hidden Markov model [17]. Equation 3 defines such a higher order model.

$$P(T_1,\cdots,T_n|D) = \big((1-\lambda_1)P(T_1|C) + \lambda_1 P(T_1|D)\big)$$
$$\prod_{i=2}^{n}\big((1-\lambda_i-\mu_i)P(T_i|C) + \lambda_i P(T_i|D) + \mu_i P(T_i|T_{i-1},D)\big) \tag{3}$$

In this case, the importance of a term T_i might be 0 (unimportant as before), 1 (important as before) and 2 (the query term T_i is generated from the bigram model). Similar to Equation 2, μ_i is used instead of $P(I_i\!=\!2)$ and $P(T_i|T_{i-1},D)$ instead of $P(T_i|T_{i-1}, I_i\!=\!2, D)$. The conditional dependencies now include dependencies between query terms. This is visualized in the Bayesian network of Figure 2.

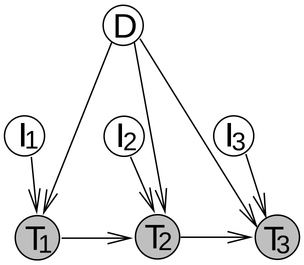

Figure 2: Graphical model of dependence relations between query terms

In such a generalized model, we have both stop words and stop bigrams. Consider for example the query `last will of Alfred Nobel`. It does not make sense to search for all possible bigrams in the document collection, because this might take to much system resources. Based on the term statistics, the system might e.g. decide that the bigrams "last will"[4], "will of" and "of Alfred" are all 'stop bigrams', that is, $\mu_i = 0$ for these bigrams. The system might decide that "of" is a stop word as well, that is, $\lambda_i = 0$ and $\mu_i = 0$ for this word.

Similarly, users might override the system's decisions if they are not satisfied with the search results. For instance, the query `"last will" of Alfred Nobel` indicates that "last will" should not be a stop bigram ($\mu_i > 0$); and the query `+"last will" of Alfred Nobel` indicates that "last will" is a mandatory bigram, that is, $\mu_i = 1$ and $\lambda_i = 0$ for this bigram.

Note that the use of single or double quotes to mark phrases is standard practice in many information retrieval systems, for instance web search engines like Google and AltaVista, and for instance commercial online services like those provided by Dialog and LexisNexis.

5. CONCLUSION

In this paper we extended the language modeling approach to information retrieval by explicitly modeling the importance of a term. By doing so, we are able to define complex models in which we might decide on a rather ad-hoc basis that some words are stop words, others are mandatory words, some pairs of words should be considered as phrases, some phrases might be mandatory, etc. Any such ad-hoc decision would override the default ranking algorithm.

Decisions on stop words and stop phrases (bigrams) are typically taken by the system, based on a trade-off between search quality and search speed. Including all document representations in each search would take too much of the system's resources, so only a few representations will be considered in practice by default, including maybe one or two bigrams.

Decisions on mandatory words and phrases are typically taken by the user. A user that is not satisfied with the system's default behaviour, or a user that has some experience with the system, will be able to mark the words and phrases he/she thinks are important. In such cases it is important that the user has some idea of how the system works. The system's default behaviour should be rather transparent, e.g. by always providing coordination level rankings.

Today's statistical ranking algorithms perform quite well in empirical settings, especially algorithms motivated by the language modeling approach. However, there will always be a few exceptions to the good performance: Some queries for which the statistics are clearly wrong. There might be only one or two of those examples in test collection with 50 topics. Whether or not the user's ad hoc decisions on mandatory words and phrases will actually result in significantly better system performance in terms of precision and recall might therefore be hard to measure. Such cases are addressed by this paper.

What about the statistical ranking algorithms of the future? Now that we have identified ways to model ad hoc decisions on multiple levels of language models in order to construct complex query representations, the next step might be trying to automate these ad hoc decisions. Such an automatic ad hoc decision (which would then of course no longer be 'ad hoc') might be based on the fact that it describes the data more *parsimoniously* [23]. For instance, in a multiple level language model, some levels might be defined as models of specific topics. As such, a unigram model of general language, modified by a small number of topic-specific unigram models, each of which has parameters for only a small number of topic-specific specialist terms and phrases, would be a very parsimonious, and possibly quite powerful, model for a collection of documents...

[4] Note that "will" is a very frequent word in English.

6. REFERENCES

[1] A. Berger and J. Lafferty. Information retrieval as statistical translation. In *Proceedings of the 22nd ACM Conference on Research and Development in Information Retrieval (SIGIR'99)*, pages 222–229, 1999.

[2] H.E. Blok, D. Hiemstra, S. Choenni, F.M.G. de Jong, H.M. Blanken, and P.M.G. Apers. Predicting the cost-quality trade-off for information retrieval queries: Facilitating database design and query optimisation. In *Proceedings of the 10th International Conference on Information and Knowledge Management (CIKM'01)*, pages 207–214, 2001.

[3] P.D. Bruza and T.W.C. Huibers. Investigating aboutness axioms using information fields. In *Proceedings of the 17th ACM Conference on Research and Development in Information Retrieval (SIGIR'94)*, pages 112–121, 1994.

[4] C. Buckley and A.F. Lewit. Optimization of inverted vector searches. In *Proceedings of the 8th ACM Conference on Research and Development in Information Retrieval (SIGIR'85)*, pages 97–110, 1985.

[5] 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.

[6] W.B. Croft, D.J. Harper, D.H. Kraft, and J. Zobel, editors. *Proceedings of the 24th ACM Conference on Research and Development in Information Retrieval (SIGIR'01)*. ACM Press, 2001.

[7] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the

em-algorithm plus discussions on the paper. *Journal of the Royal Statistical Society*, 39(B):1–38, 1977.

[8] D. Hawking and P. Thistlewaite. Relevance weighting using distance between term occurrences. Technical Report TR-CS-96-08, The Australian National University, 1996. (http://cs.anu.edu.au/techreports/)

[9] D. Hiemstra. A probabilistic justification for using *tf.idf* term weighting in information retrieval. *International Journal on Digital Libraries*, 3(2):131–139, 2000.

[10] D. Hiemstra and W. Kraaij. Twenty-One at TREC-7: Ad-hoc and cross-language track. In *Proceedings of the seventh Text Retrieval Conference TREC-7*, pages 227–238. NIST Special Publication 500-242, 1998.

[11] D. Hiemstra and A.P. de Vries. Relating the new language models of information retrieval to the traditional retrieval models. Technical Report TR-CTIT-00-09, Centre for Telematics and Information Technology, 2000. (http://www.ctit.utwente.nl)

[12] F. Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, 1997.

[13] M.I. Jordan, editor. *Learning in Graphical Models*. Kluwer Academic Press, 1998.

[14] W. Kraaij, T. Westerveld, and D. Hiemstra. The importance of prior probabilities for entry page search. In *Proceedings of the 25th ACM Conference on Research and Development in Information Retrieval (SIGIR'02)*, 2002. (in this volume)

[15] J. Lafferty and C. Zhai. Document language models, query models, and risk minimization. In *Proceedings of the 24th ACM Conference on Research and Development in Information Retrieval (SIGIR'01)*, pages 111–119, 2001.

[16] S. Lawrence and C.L. Giles. Accessibility of information on the web. *Nature*, 400:107–109, 1999.

[17] D.R.H. Miller, T. Leek, and R.M. Schwartz. A hidden Markov model information retrieval system. In *Proceedings of the 22nd ACM Conference on Research and Development in Information Retrieval (SIGIR'99)*, pages 214–221, 1999.

[18] J. Muramatsu and W. Pratt. Transparent queries: Investigating users' mental models of search engines. In In *Proceedings of the 24th ACM Conference on Research and Development in Information Retrieval (SIGIR'01)*, pages 217–224, 2001.

[19] K. Ng. A maximum likelihood ratio information retrieval model. In *Proceedings of the eighth Text Retrieval Conference, TREC-8*. NIST Special Publications, 1999.

[20] M. Persin. Document filtering for fast ranking. In *Proceedings of the 7th ACM Conference on Research and Development in Information Retrieval (SIGIR'84)*, pages 339–349, 1984.

[21] J.M. Ponte and W.B. Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st ACM Conference on Research and Development in Information Retrieval (SIGIR'98)*, pages 275–281, 1998.

[22] C.J. van Rijsbergen. *Information Retrieval, second edition*. Butterworths, 1979. (http://www.dcs.gla.ac.uk/Keith/Preface.html)

[23] S.E. Robertson and D. Hiemstra. Language models and probability of relevance, In *Proceedings of the Workshop on Language Modeling and Information Retrieval*, pages 21–25, 2001.

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

[25] S.E. Robertson and S. Walker. Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th ACM Conference on Research and Development in Information Retrieval (SIGIR'94)*, pages 232–241, 1994.

[26] J.J. Rocchio. Relevance feedback in information retrieval. In G. Salton, editor, *The Smart Retrieval System: Experiments in Automatic Document Processing*, pages 313–323. Prentice Hall, 1971.

[27] D.E. Rose and C. Stevens. V-twin: A lightweight engine for interactive use. In E.M. Voorhees and D.K. Harman, editors, *Proceedings of the 5th Text Retrieval Conference TREC-5*, pages 279–290. NIST Special Publication 500-238, 1996.

[28] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988.

[29] F. Song and W.B. Croft. A general language model for information retrieval. In *Proceedings of the 8th International Conference on Information and Knowledge Management (CIKM'99)*, pages 316–321, 1999.

[30] K. Sparck-Jones. A statistical interpretation of term specifity and its application in retrieval. *Journal of Documentation*, 28(1):11–20, 1972.

[31] R. Wilkinson, J. Zobel, and R. Sacks-Davis. Similarity measures for short queries. In D.K. Harman, editor, *Proceedings of the 4th Text Retrieval Conference TREC-4*, pages 277–286. NIST Special Publication 500-236, 1995.

[32] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th ACM Conference on Research and Development in Information Retrieval (SIGIR'01)*, pages 334–342, 2001.