A New Probabilistic Model of Text Classification and Retrieval

Tom Kalt
University of Massachusetts
kalt@cs.umass.edu

preliminary draft of January 25, 1996

Abstract

This paper introduces the multinomial model of text classification and retrieval. One important feature of the model is that the *tf* statistic, which usually appears in probabilistic IR models as a heuristic, is an integral part of the model. Another is that the variable length of documents is accounted for, without either making a uniform length assumption or using length normalization. The multinomial model employs independence assumptions which are similar to assumptions made in previous probabilistic models, particularly the binary independence model and the 2-Poisson model. The use of simulation to study the model is described. Performance of the model is evaluated on the TREC-3 routing task. Results are compared with the binary independence model and with the simulation studies.

1 Introduction

Probabilistic models of information retrieval have usually focused on binary features, such as the presence or absence of an index term in a document. They have had limited success at directly incorporating the number of occurrences of a term, although the *tf* statistic has long been recognized as a very important component of practical retrieval functions. In fact, *tf.idf* weights are perhaps the biggest success story in IR. The theory developed here takes a different point of departure from most previous approaches, namely, the process by which text is generated. Otherwise, the development is similar to that of most other probabilistic models. The discriminant function for the model and its derivation turn out to be relatively simple.

We begin by assuming that text is generated by a stochastic process. A source outputs a sequence of symbols, conceptually infinite, which for our purposes are words. A document is a sample, that is, a particular sequence, taken from some source. The statistical properties of documents are therefore determined by the nature of the stochastic processes which generated them. Furthermore, a class of documents is considered to be generated by a single source. Discrimination between document classes is based on computing the probability that a document was generated by a particular source.

By specifying different kinds of stochastic processes, this framework gives rise to a family of models. The simplest process is the discrete memoryless source, which leads to the multinomial model. A discrete memoryless source generates symbols with fixed probability; that is, the probability that the next word generated will be a particular term depends only on the term, and is independent of anything that was generated previously. An example of a more complex process is a Markov chain in which each term is conditionally dependent on the previous n terms generated, for some fixed window size. The "upgrade path" for the multinomial model thus involves relaxing the independence assumptions which have always been an issue in probabilistic IR. We hope in this way to incorporate the notion of conditional dependence on context into our models. Also, giving the source a limited memory will allow us to work with compound features such as phrases. The utility and feasibility of models based on more complex stochastic processes will depend on their mathematical tractability, as well as the feasibility of estimating parameters for them.

The multinomial model is thus a first-order model, which makes the simplest possible assumptions. In this respect, it is comparable to the binary independence model [5].

2 The Multinomial Model

2.1 Text Generation

Formally, we begin with a finite lexicon $\mathcal{L}$, each element of which is a term. Following the common practice in IR, we consider words to be the basic units of text, ignoring punctuation, etc. Two documents consisting of the same sequence of terms are considered identical.

A discrete memoryless source S emits a sequence of elements of $\mathcal{L}$ according to a probability distribution $\{\alpha_i\}$. A source is completely specified by a lexicon and a distribution. Each term t_i in the lexicon has a fixed probability α_i of being generated at each time step. Thus $\{\alpha_i\}$ is subject to the constraint $\sum_i \alpha_i = 1$. This process is the same as repeatedly rolling a die with $\mid \mathcal{L} \mid$ faces, where α_i is the probability that the die will come up indicating term t_i.

A discrete memoryless source can also be thought of as a collection of Poisson processes. A stochastic process consisting of a separate Poisson process for each term t_i in the lexicon, each with Poisson rate α_i, is equivalent to the discrete memoryless source described above. To see this, suppose that each Poisson process emits words at real-valued times, and suppose we order all the words emitted by their times of emission. The resulting sequence is equivalent to the output of a discrete memoryless source. The probability that any particular term in the output is t_i is its Poisson rate, and each token is independent of its predecessors. The present theory is thus related to the 2-Poisson model [1], a model of term distribution in which the *tf* statistic plays an important role.

For any class of documents C, we posit a source $\mathcal{S}$, which generates that class. For example, in an IR application, we would say that the relevant set is a sample of documents taken from one source, and the rest of the collection is a sample taken from another. A given document could have been generated by many different sources; but it is much more likely to have been generated by some than by others. Sources are thus an abstraction used to encode the statistical properties of a document class.

We don't require that all documents generated by $\mathcal{S}$ are in C, nor that documents in C can only be generated by $\mathcal{S}$. Nor do we require $\mathcal{S}$ to be memoryless; it could be a more complex process. $\mathcal{S}$ itself is in general inaccessible, and the primary information we have about it is obtained from a sample. For any finite sample $\{D_i\}$ of documents from C, there is a maximum likelihood memoryless source S which generates the sample. This source is one which maximizes $P(\{D_i\} \mid S)$, the probability that the given sample will be generated by the source. The distribution $\{\alpha_i\}$ for the maximum likelihood source is obtained by dividing the number of times each term t_i occurs by the size of the sample.

A good analogy is modelling a coin-flipping process with a binomial random variable. The real-world process could be very complicated. It could even have strong deviations from independence, as long as the effects are local, that is, there is some number of flips after which the effects disappear. A binomial random variable is a good model for such a process, and the maximum likelihood estimate of the probability of heads is given by the number of heads in a large sample divided by the sample size.

2.2 Document Representation

A document is a sequence of n tokens taken from a particular source. We do not explicitly model the details of the sampling process, such as how the beginning and end of a document are chosen. We stipulate only that a document is a particular sequence generated by a single source; this rules out taking every other word, for example. This restriction is not necessary for the multinomial model. However, in future elaborations of the model which will give the source some memory, the actual sequence of symbols will matter.

Having said what a document is, we now come to the matter of representation. The representation we will use is the vector $\mathbf{x}$ where the component x_i indicates the number of times term t_i occurs in the document. By comparison, the component x_i of the binary representation vector $\mathbf{x}$ indicates the presence or absence of term t_i. This choice turns out to be pivotal, because it leads to the multinomial distribution.

2.3 Discrimination

We now derive the discriminant function for the multinomial model. The following analysis is common to classification and retrieval, although the motivation is slightly different for the two problems. In IR, the probability ranking principle says that the optimal way to present documents to the user is to *rank* them by decreasing order of $P(R_Q \mid D)$, the probability of relevance to the query. The text classification problem is to *decide* whether a document D belongs to a class C_1 or its complement C_2. C_1 is analogous to the relevant set R_Q. The optimal way to decide whether a document belongs to a class is given by Bayes' decision rule: decide C_1 if $P(C_1 \mid D) > P(C_2 \mid D)$, otherwise decide C_2. So for both IR and classification, our initial goal is to compute $P(C \mid D)$.

In practice, we condition not on D, but on its representation $\mathbf{x}$. The desired probability may be computed using Bayesian inversion. For simplicity, we use the odds form of Bayes' theorem:

$$O(C_1 \mid \mathbf{x}) = L(\mathbf{x} \mid C_1) O(C_1) \tag{1}$$

where L is the conditional likelihood ratio

$$L(\mathbf{x} \mid C_1) = \frac{P(\mathbf{x} \mid C_1)}{P(\mathbf{x} \mid C_2)} \tag{2}$$

and $O(C_1)$ is the prior odds of C_1, which is

$$O(C_1) = \frac{P(C_1)}{P(C_2)} \tag{3}$$

For IR, it is sufficient to rank by $L(\mathbf{x} \mid C_1)$ or $\log(L(\mathbf{x} \mid C_1)$, since $O(C_1)$ doesn't depend on $\mathbf{x}$, and $L(\mathbf{x} \mid C_1)$ is a monotonic function of $P(C_1 \mid \mathbf{x})$. For classification, we additionally need to estimate the prior odds $O(C_1)$.

Now suppose we define S_1 as the source which emits documents in C_1; likewise, documents in C_2 come from source S_2. These sources are characterized by term probability distributions $\{\alpha_{1i}\}$ and $\{\alpha_{2i}\}$. In effect, what we are saying is that we intend to discriminate between C_1 and C_2 based on the difference in term frequencies in the two classes.

We have chosen the representation such that the probability of observing a particular $\mathbf{x}$ from a source S_1 is given by the multinomial distribution:

$$P(\mathbf{x} \mid C_1) = P(\mathbf{x} \mid S_1) = n! \prod_{t_i \in \mathcal{L}} \frac{\alpha_{1i}^{x_i}}{x_i!} \tag{4}$$

where n is the length of the document in words, and x_i is the number of occurrences of term t_i. By itself, this formula is not very useful. It is easy to see that $P(\mathbf{x} \mid C_1)$ is always extremely small. However, this difficulty is resolved when we compute the likelihood ratio. Substituting Eq. 4 into Eq. 2, we get

$$L(\mathbf{x} \mid C_1) = \frac{n! \prod_{t_i \in \mathcal{L}} \frac{\alpha_{1i}^{x_i}}{x_i!}}{n! \prod_{t_i \in \mathcal{L}} \frac{\alpha_{2i}^{x_i}}{x_i!}} = \prod_{t_i \in \mathcal{L}} \left(\frac{\alpha_{1i}}{\alpha_{2i}}\right)^{x_i}$$

and taking logs, we have

$$g(\mathbf{x}) = \log L(\mathbf{x} \mid C_1) = \sum_{t_i \in \mathcal{L}} x_i \log\left(\frac{\alpha_{1i}}{\alpha_{2i}}\right) \tag{5}$$

This discriminant function $g(\mathbf{x})$ may be used for ranking in IR, or for decision in classification. In this paper, we use routing as our test environment, and we assume that ample training data is available for estimating α_{1i} and α_{2i}. Also, our discriminant functions are purely inductive; we don't include a component derived from an original query or a topic description.

2.4 Discussion

This linear discriminant function $g(\mathbf{x})$ has a term for each t_i in the lexicon. It is symmetrical with respect to the two classes; the roles can be reversed by exchanging α_1 and α_2. Each term in the sum is the product of a part that depends on the document (x_i) and a part that depends on the classification problem ($\log(\alpha_1/\alpha_2)$). These factors correspond to document weights and query weights in IR. Since the query weight can be negative, $g(\mathbf{x})$ can be positive or negative. For arbitrarily large documents, $\mid g(\mathbf{x}) \mid$ can be arbitrarily large.

For every term that occurs in a document, $g(\mathbf{x})$ is increased if the term has a higher rate in S_1 than in S_2, or decreased if the converse is true. If the term is equally common in C_1 and C_2, then the presence of that term contributes nothing to $g(\mathbf{x})$, no matter what x_i is. There is no contribution to $g(\mathbf{x})$ for any term that is absent from the document.

An important characteristic of $g(\mathbf{x})$ is that it incorporates x_i directly. In IR, x_i is usually denoted by tf_i, the "within-document frequency" of t_i.[1] Although tf is known to be extremely important in retrieval functions, it has usually come into probabilistic models of retrieval in a more or less *ad hoc* manner. It appears here as a direct consequence of the document representation.

Another important feature of this discriminant function is the way it incorporates document length. Previous models have usually assumed that documents were roughly the same length. Most probability estimation schemes for IR include some form of document length normalization. Here, the multinomial distribution gives us the probability for a document of any length. So length is accounted for, without having either to make the dubious assumption of uniform document length (see Figure 3), or perform a length normalization.

Consider how this function estimates $g(\mathbf{x})$ for two documents $\mathbf{x}$ and $\mathbf{x}\prime$, where $\mathbf{x}\prime$ consists of two repetitions of $\mathbf{x}$. Rather than normalizing the difference away, this discriminant function finds more evidence for C_1 in $\mathbf{x}\prime$ than in $\mathbf{x}$, specifically $g(\mathbf{x}\prime) = g(\mathbf{2x}) = 2g(\mathbf{x})$.

2.5 Independence Assumptions

In both the binary independence model and the present model, independence is not something we really believe, except as a very coarse approximation. We make the assumptions for mathematical convenience; if they become an obstacle, we try to relax them (although this has not proved easy for the binary independence model; see [9]).

Cooper [2] describes the assumption of the binary independence model as "linked dependence", which he argues is considerably weaker than a true independence assumption. That argument does not apply to the multinomial model. For each of the two discrete memoryless sources, the symbols generated are assumed to be strictly independent. Perhaps more importantly, the present model makes an assumption of independence at a finer level of granularity. In the binary independence model, the events assumed independent are the occurrence or nonoccurrence of terms in a document. In the multinomial model, each time a word is generated by the source, that event is independent of all other such events. Thus the multinomial model's independence assumption is stronger than that of the binary independence model.

Much could be said about the relationship between the multinomial model and the 2-Poisson model, which can only be sketched here. In the 2-Poisson model, each term which has a 2-Poisson distribution partitions the collection: in some documents, the term has a high rate; in the rest, it is low. In the multinomial model, the partition is given by the classification problem. Some terms will be found to have a high rate in one class and a low rate in the other. For other terms, the rates will not be appreciably different.

2.6 Document Length and Term Selection

In order to understand better how document length is accounted for in this model, it is interesting to consider the term selection problem. The model specifies that $g(\mathbf{x})$ is a sum over all terms. In practice, some subset of the lexicon must be chosen. This is necessary not only for efficient query processing, but also because we can't always estimate α accurately. One might be tempted to simply eliminate the deselected terms from the sum, by making the assumption $\alpha_1 = \alpha_2$ for those terms. This is the approach used in the binary independence model, where it is perfectly appropriate. We can't do that here, because it would violate the model.[2] Instead, the model requires that all terms not explicitly included in the sum should be conflated into a single "default term", with its own Poisson rate. Thus at each time step, the source emits either one of the selected terms or the default term. If there are n selected terms, the model becomes an $n+1$-outcome multinomial. The length of a document is therefore preserved, regardless of the number of terms selected for inclusion in $g(\mathbf{x})$. The revised formula is

[1] We prefer to call x_i the term's count, or its number of occurrences, rather than its frequency, which implies normalization; normalized counts will be referred to as rates (e.g., Poisson rates).

[2] Consider what would happen if only one term were chosen. Then the model would be a binomial.

$$g(\mathbf{x}) = \sum_{t_i \in \mathcal{T}} x_i \log\left(\frac{\alpha_{1i}}{\alpha_{2i}}\right) + x_d \log\left(\frac{\alpha_{1d}}{\alpha_{2d}}\right) \quad (6)$$

where $\mathcal{T}$ is the set of selected terms, x_d is the number of occurrences of the default term, and α_{1d} and α_{2d} are the Poisson rates of the default term in classes C_1 and C_2 respectively. As before, we require that

$$\sum_{t_i \in \mathcal{T}} \alpha_i + \alpha_d = 1 \quad (7)$$

Many issues related to term selection are still unresolved. In particular, the best way to select a small subset of terms, and how to determine the number of terms to use, are unknown.

2.7 Parameter estimation

Initially, it appears we need to estimate α_{1i} and α_{2i} for all terms in the lexicon. For IR purposes, we can reduce this to the set of terms which occur in the relevant set.

The maximum likelihood estimate of α_i, using a set of training documents for a single class, is to divide the total number of occurrences of t_i in the training set by the size of the training set in words. Another approach is to compute α_i for each document, and then average the results. This gives equal weight to each document, rather than weighting documents by their length, as the first approach does. We adopted this approach for the experiments reported here (for the relevant sets only), because of concerns about the extremely skew distribution of document lengths.

The term selection problem is eased by eliminating terms for which $\frac{\alpha_{1i}}{\alpha_{2i}}$ is close to 1. In the work reported here, we used a significance test on the difference rather than the ratio. Treating each α as a mean, the well-know z test computes the probability that the observed difference in means could have arisen by chance.

3 Simulation studies

One of the virtues of probabilistic models is that it is often easy to simulate them. Properties of the models which could not be derived analytically can then be determined empirically by running the simulation. For example, it is reasonable to ask what the expected performance of a model would be, once the parameters it uses have been determined. We would also like our model to predict how performance depends on the method of term selection and the number of terms used. These questions can be explored through simulation.

Simulations of both the multinomial model and the binary independence model were implemented. We simulated performance on the TREC-3 routing task [3], using disk 1 for training data. We computed term statistics from the relevant and non-relevant document sets for each topic. All terms occurring in the relevant set were considered. All documents not in the relevant set were considered non-relevant. Terms were selected by the same method used in the actual performance studies reported below, and sixteen terms per query were used.

The design of the simulation programs is straightforward. Both are small standalone programs; no retrieval engine is needed. The multinomial model simulation will be described. The binary independence version is similar.

The input required for simulating performance on a single topic is the set of selected terms, each with its Poisson rate in the relevant and non-relevant sets (R and NR) respectively. Also, the sizes of R and NR are required. In the first phase of processing, distributions of $g(\mathbf{x})$ in R and NR are computed. The second phase is to compute an estimated recall-precision table from the distributions. This outline is repeated in more detail in Figure 3.

One detail that deserves mention is the way document length is determined. Since the value of the discriminant function depends strongly on document length, it is important to model the distribution of lengths accurately. We do this by picking a length at random from a large sample of actual document lengths taken from disk 1. In the simulation, therefore, document lengths have approximately the same distribution as the documents of the training set.

Figure 3 shows the distribution of lengths for TREC Disk 1. The median length is 104 words (after stopword removal). Not shown is the very heavy tail. Almost two percent of the documents have lengths over 1000 words. The variance of document lengths is a major source of variance in $g(\mathbf{x})$ for the multino-

```
SIMULATE
    for each query
        for each class $C_i$
            SIMULATE_SOURCE
            for $j = 1$ to $samplesize$
                pick a length $l$ from disk 1 lengths
                get a sequence of $l$ words from source
                compute its representation $\mathbf{x}$
                $G_i[j] \leftarrow g(\mathbf{x})$
        EVALUATE

SIMULATE_SOURCE
    for each $t_i$ in query
        $E(n_i) \leftarrow \alpha_i \times source\_size$
        put $E(n_i)$ copies of $t_i$ in source array
    scramble source array

EVALUATE
    sort each array $G_i$
    for each standard recall point
        compute precision
```

Figure 1: Pseudocode for multinomial model simulation

mial model simulation. For the binary independence model, this effect does not show up in the simulation, since documents are assumed to have uniform length. However, it does have an effect in the real world, since any term is more likely to occur in a longer document.

It should also be noted that in the simulation of the binary independence model, the independence assumption is stronger than "linked dependence". That is

$$P(t_i, t_j \mid R) = P(t_i \mid R)P(t_j \mid R) \qquad (8)$$

and

$$P(t_i, t_j \mid \not R) = P(t_i \mid \not R)P(t_j \mid \not R) \qquad (9)$$

We have not yet devised a method of modelling linked dependence directly. It would of course be quite interesting to know how linked dependence compares to true independence under simulation.

The expected performance for the two models is shown in Table 1. The comparison is not as informative as we had originally hoped, since both models are wildly optimistic, predicting almost perfect performance. So we cannot say on the basis of simulation studies which model we would expect to perform better; nor can we believe the predicted performance. Because expected precision is already saturated, the simulations are not likely to be a good way to explore term selection methods. We hope this situation will change as the models improve.

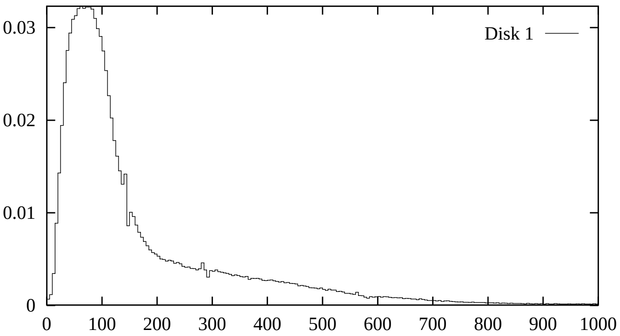

Figure 2: Distribution of document lengths in TREC data (disk 1)

The simulations predict the performance their discriminant functions would give, for the parameters of the TREC-3 routing topics, if terms were really distributed as the models assume. The wide gap between predicted and actual performance reflects the difference between the statistical properties of our simple models and those of real documents.

4 Performance Evaluation

A program to retrieve documents using the multinomial model was implemented by modifying the Inquery retrieval engine. For comparison, we also implemented the binary independence model. Both implementations used Inquery's lexical analysis, consisting of case folding, stemming, and stopword removal.

The programs were evaluated using a variant of the TREC-3 routing task. Our experiments used the documents of disk 1 for training and disk 2 for evaluation. The discriminant function was encoded as a query consisting of a set of terms with weights.

A number of runs were done to compare term selection methods and to vary query length. The term selection methods consisted of ranking terms by various statistics, and picking the top n. The best performer for both models, which was used for

Recall	Precision	
	BIM	MM
0	100.0	100.0
10	100.0	100.0
20	100.0	100.0
30	99.9	100.0
40	99.9	100.0
50	99.7	100.0
60	97.6	100.0
70	96.1	99.9
80	94.0	96.1
90	89.0	79.0
100	24.5	0.1
avg	91.0	88.6

Table 1: Simulated performance of binary independence and multinomial models on the TREC-3 Routing task

Recall	Precision	
	BIM	MM
0	81.5	60.0
10	56.1	40.6
20	43.4	35.8
30	37.2	29.8
40	32.0	25.2
50	27.5	21.4
60	23.0	17.1
70	18.7	13.6
80	14.5	9.5
90	10.6	6.1
100	4.1	1.4
avg	31.7	23.7

Table 2: Actual performance of binary independence and multinomial models on TREC-3 Routing

the results reported here, was an *ad hoc* function, $ratio \times rdf$, where $ratio = \frac{\alpha_{1i}}{\alpha_{2i}}$ and rdf is the number of relevant documents containing t_i. The runs reported in Table 2 used 16 terms per query.

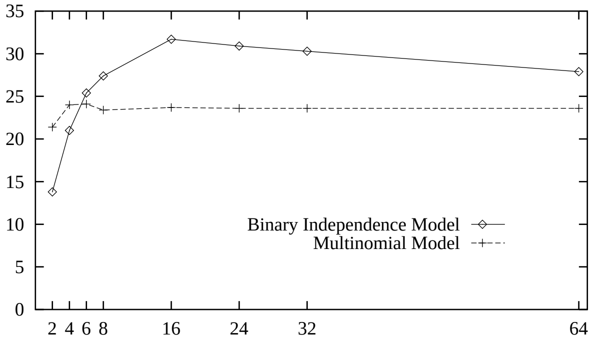

Figure 3: Average precision vs. number of terms per query

Figure 3 shows that the binary independence model is more sensitive to query length than the multinomial model. Overall, however, the results show that performance of the multinomial model on this task is substantially lower than that of the binary independence model. We hypothesize that this is primarily due to the stronger independence assumptions of the multinomial model. It can be argued that the binary statistics used in the binary independence model are more robust than the ones used in the multinomial model. The violations of the multinomial model's independence assumption may well be more detrimental than the corresponding violations for the binary independence model.

In particular, there is one effect to which the multinomial model is subject which has no counterpart in the binary independence model. It is likely that many terms, especially the kind that are likely to be good discriminators, are strongly dependent on their own occurrence. That is, knowing that t_i has occurred once in a document, we should be much less surprised to see it a second time than we were the first time. Therefore, if we treat each occurrence of such a term as independent, we are likely to seriously overestimate the importance of that term. This consideration is related to the use in many IR systems of a function of *tf* which grows more slowly than *tf* itself. Some of these are:

$$\begin{array}{ll} (0.5 + 0.5 \times tf/maxtf) & [8] \\ log(tf + 1) & \\ tf/(tf + const) & [7] \end{array}$$

Ultimately, we would like any modification to *tf* to come from changes in the model; and if the model changes, the entire discriminant function is likely to be affected. As noted in the introduction, addressing the term-dependence problem will be a high priority in future versions of the theory.

5 Related Work

The 2-Poisson model [1] introduced the idea of multiple Poisson distributions. That work was primarily concerned with the problem of automatic indexing, that is, identifying a set of terms by which a document could be represented. Documents were assumed to be of uniform length, and the theory was developed using Poisson means rather than Poisson rates, where the mean is the rate times document length. The TPI model [6] is a combination of the binary independence model and the 2-Poisson model. A previous approach which extended the binary independence model to cover multiple term occurrences is described in [11] and [12]. Another approach in which text is generated by a stochastic process is [4], which employs a hidden Markov model.

6 Conclusions and future work

The multinomial model gives a simple account of how *tf* can come into probabilistic models, through modeling the generation of text by a stochastic process. It also takes document length into account in a new and well-justified way. Actual performance of the model on the TREC-3 routing task is too low for the model to be used in practical applications in its present form. The independence assumptions which give the model simplicity appear to be its weakness as well. Nevertheless, this work sheds some light on the role of *tf* and document length in probabilistic IR. It also prepares the way for better models based on more complex stochastic processes.

Simulation of probabilistic models is shown to be a fruitful way of extending probabilistic theories beyond the realm of analytic technique. Simulation shows a huge gap between expected and actual performance for both the binary independence model and the multinomial model.

The role of conditional dependence will be a central focus of future work. Specifically, we want to investigate models in which the probability of generating a term may depend on the local context in various ways.

References

[1] Bookstein, A. and D. R. Swanson, Probabilistic models for automatic indexing. *Journal of the ASIS* 25(5):312-319, 1974

[2] Cooper, William S. Inconsistencies and misnomers in probabilistic IR. *Proc. ACM SIGIR Conference on R & D in Information Retrieval* pp. 57-61, 1991

[3] Harman, D. Overview of the third text retrieval conference (TREC-3). *The Third Text REtrieval Conference (TREC-3)* National Institute of Standards and Technology. 1995

[4] Mittendorf, Elke and Peter Schäuble. Document and passage retrieval based on hidden Markov models. *Proc. ACM SIGIR Conference on R & D in Information Retrieval* pp. 318-327, 1994

[5] Robertson, S. E. and K. Sparck-Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science* 27:129-46, 1976

[6] Robertson, S. E., C. J. van Rijsbergen, M. F. Porter. Probabilistic models of indexing and searching. In Oddy, R. N., *et al.*, *Information Retrieval Research* pp 35-56, Butterworths. 1981

[7] Robertson, S. E. and S. Walker, Some simple effective approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval. *Proc. ACM SIGIR Conference on R & D in Information Retrieval* pp. 232-241, 1994

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

[9] van Rijsbergen, C. J. A theoretical basis for the use of co-occurrence data in information retrieval. *Journal of Documentation* 33(2):106-119, 1977

[10] van Rijsbergen, C. J. *Information Retrieval.* Butterworths. 1979

[11] Yu, C. T. and T. C. Lee, Non-Binary Independence Model. *Proc. ACM SIGIR Conference on R & D in Information Retrieval* pp. 265-268, 1986

[12] Yu, C. T. and H. Mizuno, Two learning schemes in information retrieval. *Proc. ACM SIGIR Conference on R & D in Information Retrieval* pp. 201-218, 1988