Testing the Maximum Entropy Principle for Information Retrieval

Paul B. Kantor and Jung Jin Lee*
Alexandria Project Laboratory (APLab), School of Communication, Information and Library Studies (SCILS), Rutgers University, 4 Huntington Street, New Brunswick, NJ 08903. E-mail: kantor@scils.rutgers.edu

A probabilistic information retrieval method using the Maximum Entropy Principle (MEP) was proposed by Cooper and Huizinga (1982). Several refinements of the MEP for information retrieval have been proposed by Kantor and Lee (1986, 1991), but the MEP has not been evaluated in any large database. This article examines the MEP retrieval method using the TREC5 database. The MEP is evaluated by several tests and compared with a "naive ordering method" and "lexicographic ordering method." The MEP does not provide any startling improvement, and it works reasonably well only in the case of a small number of keys and a relatively small collection.

1. Introduction

Some time ago, Cooper and Huizinga (1982) and Cooper (1983) proposed that the Maximum Entropy Principle (MEP) could be applied to the problem of information retrieval (IR). Specifically, they conjectured that information about the ''value'' of individual terms, in retrieving documents relevant to a specific query, could be combined with the MEP to estimate the effectiveness of term combinations in retrieval. Shortly thereafter, Kantor (1984) developed a rigorous formulation of the MEP for IR, in terms of a set of coupled non-linear equations for certain parameters λ_t which determine the effectiveness of term combinations. This work has had almost no impact, although Kantor and Lee (1986, 1991) did provide one ''toy calculation'' showing that the MEP ''behaves sensibly'' in the situation where terms co-occur very frequently. In that case, they showed, the MEP correctly avoids the conclusion that the occurrence of both terms together is especially significant for retrieval purposes.

What has remained open are the questions of whether the MEP is (a) true, for reasonable collections of documents and sets of questions or ''bi-corpora'' (this term is adopted from Kantor, 1994), or (b) useful for information retrieval, even if it is not true. In the present article, we examine these questions using the TREC5 (Harman, 1996, in press) bi-corpus. In particular, we have concentrated on the 50 routing topics, and on documents which have either been judged as to relevance, or are contained in 0.1%, 1%, and 10% random samples of the TREC5 documents.

There are several ways in which the MEP might be found to apply. The Strong MEP asserts that the actual distribution of relevant and non-relevant documents across the collection corresponds, as a probability distribution, to the distribution generated by the MEP. The Rank MEP asserts (more weakly) that the MEP orders the possible term combinations in the most effective order for optimal retrieval. That is, under the Rank MEP, it is sufficient that each pair of term combinations be ordered correctly, even if the actual distribution of relevant and non-relevant documents is not described correctly. (We note, parenthetically, that this form of the MEP is adequate for the routing task, but is not adequate for solving the so-called filtering task; Lewis, 1996). Finally, we consider the Score MEP. Under the Score MEP, it is sufficient that the expected average precision score for the results produced by the MEP exceed the score produced by several variant ''null hypotheses'' concerning the distribution of relevant and non-relevant documents.

The MEP does not work as well as we might hope. Possible explanations are considered in the conclusion.

2. Maximum Entropy Principle

We shall use the term ''documents'' to refer in general to any retrievable report or item that might contain further information relevant to the problem at hand. We first illustrate the maximum entropy principle by using two index terms denoted by A and B, and then state our results for general case. Each member of the entire set of ''documents'' may be indexed by neither, either, or both of two terms A and B. We will call a subset in which every Boolean variable takes a specific value, an atom. We

*Permanent Address: Soong Sil University, Seoul, Korea.

Received May 6, 1997; revised June 13, 1997; accepted June 13, 1997.

TABLE 1. Notation used for four Boolean atoms.

Atom no.	Boolean atom	Probability of Not relevant $p(i, 0)$	Probability of Relevant $p(i, 1)$	Fraction of documents in the atom $f_i = p(i, 0) + p(i, 1)$
1	$\bar{A}\bar{B}$	$p(1, 0)$	$p(1, 1)$	$f_1 = p(1, 0) + p(1, 1)$
2	$\bar{A}B$	$p(2, 0)$	$p(2, 1)$	$f_2 = p(2, 0) + p(2, 1)$
3	$A\bar{B}$	$p(3, 0)$	$p(3, 1)$	$f_3 = p(3, 0) + p(3, 1)$
4	AB	$p(4, 0)$	$p(4, 1)$	$f_4 = p(4, 0) + p(4, 1)$

represent the four possible Boolean atoms of the entire set of ''documents,'' D, using the notations of set operations as follows:

$$D = \bar{A}\bar{B} \cup \bar{A}B \cup A\bar{B} \cup AB.$$

Note that the expression as $\bar{A}B$ is used to represent both the logical combination of terms and the set of documents indexed by that logical combination. Let f_i, $i = 1, 2, 3, 4$, be the fraction of all documents lying in each Boolean atom. For a given query, the atoms may be further subdivided into documents which are relevant, and those which are not. We make the standard assumptions that every document is either relevant or not, and that all relevant documents are equally ''good.'' Suppose that the value of a document is either 0 or 1 which represent ''not relevant'' or ''relevant,'' respectively. Let $p(i, v)$ denote the probability that a document has value v and falls in the Boolean atom i where $i = 1, 2, 3, 4$ and $v = 0, 1$. The situation may be described by Table 1.

Since $p(i, v)$ is the joint probability and f_i is the marginal probability of each Boolean atom, we have the following probability constraints

$$\sum_{i=1}^{4} \sum_{v=0}^{1} p(i, v) = 1$$

$$p(i, 0) + p(i, 1) = f_i \quad i = 1, 2, 3, 4$$

$$p(i, v) \geq 0, \quad i = 1, 2, 3, 4 \quad \text{and} \quad v = 0, 1. \quad (1)$$

We want to provide an ordering of the Boolean atoms, in response to a user's request, by estimating the conditional probability of ''relevance'' given that the document falls in a particular atom. Note that the conditional probability that, for example, an item in Boolean atom 3 be relevant, is $p(3, 1)/f_3$. Since the fraction f_i lying in each Boolean atom can usually be determined using a computer, the question is how to estimate the joint probability $p(i, v)$, or ''relevance decomposition'' of the documents in each atom. If we know all the $p(i, v)$, then the Boolean atoms can be ranked by the order of the conditional probability $p(i, 1)/f_i$, $i = 2, 3, 4$. (Since we are not interested in the documents which have neither of two index terms A and B, the Boolean atom 1, $\bar{A}\bar{B}$, is removed from consideration.) The atom which has the highest conditional probability will be the first candidate for ''relevant'' information retrieval.

It is impossible for an expert (or an expert system) to estimate all the $p(i, v)$. However, an expert might be able to provide an opinion in the form ''the chance that documents indexed by the term A (or B) are relevant is V_A (or V_B) and the overall chance of a relevant document is V_R.'' Note that V_A, V_B, and V_R can be represented using $p(i, v)$ and f_i as follows:

$$V_A = \frac{p(3, 1) + p(4, 1)}{f_3 + f_4}$$

$$V_B = \frac{p(2, 1) + p(4, 1)}{f_2 + f_4}$$

$$V_R = p(1, 1) + p(2, 1) + p(3, 1) + p(4, 1). \quad (2)$$

The expert estimates of V_A, V_B, and V_R provide partial information on the data structure and can be used to estimate the $p(i, v)$ using the MEP. The resulting constrained optimization problem is to maximize the entropy function of the probabilities $p(i, v)$ subject to V_A, V_B, V_R and the probability constraints as follows:

Find $p(i, v)$, $i = 1, 2, 3, 4$ and $v = 0, 1$ which maximizes

$$-\sum_{i=1}^{4} \sum_{v=0}^{1} p(i, v) \ln p(i, v) \quad (3)$$

subject to

$$p(3, 1) + p(4, 1) = (f_3 + f_4)V_A$$

$$p(2, 1) + p(4, 1) = (f_2 + f_4)V_B$$

$$p(1, 1) + p(2, 1) + p(3, 1) + p(4, 1) = V_R$$

$$\sum_{i=1}^{4} \sum_{v=0}^{1} p(i, v) = 1$$

$$p(i, 0) + p(i, 1) = f_i \quad i = 1, 2, 3, 4$$

$$p(i, v) \geq 0 \quad i = 1, 2, 3, 4 \quad \text{and} \quad v = 0, 1.$$

This is a non-linear programming problem which has linear constraints. If we use the conditional probability q_i, the probability of relevance in i^{th} atom, $q_i = p(i, 1)/f_i$,

$i = 1, 2, 3, 4$, then the MEP optimization (Equation 3) problem can be written as follows:

Find q_i, $i = 1, 2, 3, 4$ which maximizes

$$-\sum_{i=1}^{4} \{f_i q_i \ln(f_i q_i) + f_i(1 - q_i)\ln(f_i(1 - q_i))\} \quad (4)$$

subject to

$$f_3 q_3 + f_4 q_4 = (f_3 + f_4)V_A$$
$$f_2 q_2 + f_4 q_4 = (f_2 + f_4)V_B$$
$$f_1 q_1 + f_2 q_2 + f_3 q_3 + f_4 q_4 = V_R$$
$$0 \le q_i \le 1, \quad i = 1, 2, 3, 4.$$

If we have the solution of the optimization problem (Equation 4), then the Boolean atom which has the highest value of q_i, $i = 2, 3, 4$, will be the best candidate for ''relevant'' information retrieval. If there are K index terms, then there are 2^K Boolean atoms and the MEP optimization problem (Equation 4) is generalized as follows (Lee & Kantor, 1991):

Find q_i, $i = 1, 2, \ldots, 2^K$ which maximizes

$$-\sum_{i=1}^{2^K} \{f_i q_i \ln(f_i q_i) + f_i(1 - q_i)\ln(f_i(1 - q_i))\} \quad (5)$$

subject to

$$\sum_{i \in A(k)} f_i q_i = \left(\sum_{i \in A(k)} f_i\right) V_k, \quad k = 1, 2, \ldots, K$$
$$\sum_{i=1}^{2^K} f_i q_i = V_R$$
$$0 \le q_i \le 1, \quad i = 1, 2, \ldots, 2^K$$

where V_k, $k = 1, 2, \ldots, K$, represents the probability of relevance for documents indexed by term k, V_R is the probability of relevance for all documents and $A(k)$ is the set of the Boolean atoms which are constrained by the index term k. This problem is solved in our experiments by using the GRG2 algorithm (Lasdon & Waren, 1979), in RM-Fort FORTRAN (Ryan-McFarland, 1986), running on a Pentium PC-clone.

3. The Test Collection

The test collection is drawn from the TIPSTER document collection, and the judgments of relevance were made by analysts working at National Institute of Standards and Technology (NIST) for the several TREC conferences. Details are given in Harman (1996). We used the so-called routing topics from the TREC5 conference as our set of queries. From each query some plausible index terms were selected by one of us, as content terms likely to have been specified by a user in a search of this topic. We discuss this procedure in section 7. In what follows, we use the term ''relevant'' to describe those documents which have been submitted to judges, and judged relevant. We make the assumption that *all other documents in the collection are not relevant to a given topic*. To provide a corpus of non-relevant documents, we drew random samples from the TIPSTER Volumes 1, 2, 3 (2 CD-ROMS). Any documents known to be relevant to any of the topics were excluded from these samples. We did some preliminary experiments in which documents that were judged not relevant were used as the corpus of non-relevant documents. We do not report these results here. We believe that they are quite different from the real situations, as most of those non-relevant, but judged documents were retrieved on the basis of terms which probably included the ones we have used as key terms.

The number of relevant documents for a topic varies from 6 to 516, but most of them are in the range between 100 to 200. The 0.1% random sample, the 1% random sample, and the 10% random sample are selected and considered as non-relevant documents. The total number of documents is 1,046,855, and the collected samples are 1,005, 10,533, and 104,618 respectively. The size of these samples is reasonable, given the size of the original set.

4. The Tests

Our tests are quite computationally intensive, and it was necessary to automate many of the steps. We considered several different questions. Suppose a given query is represented by K index terms. It is possible to form all the $\binom{K}{m}$ subsets of size m, $2 \le m \le K$. The total number of subsets is as follows:

$$\binom{K}{2} + \binom{K}{3} + \cdots + \binom{K}{K} = 2^K - K - 1. \quad (6)$$

One test is constructed from each subset. Thus if $K = 3$, there are 4 tests, if $K = 4$, there are 11 tests, and if $K = 5$, there are 26 tests of the MEP. For each topic, some plausible index terms, varying in number from 2 to 5 terms are selected by considering the topic statement. In our experiments, there are 523 tests of the MEP in all. Of course the results of different orders, for the same query, are not statistically independent of each other, nor are the results of the same order, involving different terms. However, we find it useful to organize the results in this way, in order to explore the dependency of the effectiveness of the MEP on the complexity of the terms' structures considered. The test collection is indexed using *mgquery* (Witten, Moffat, & Bell, 1994). For each subset of size m, search command files are generated. A typical

TABLE 2. An example of mgquery command file.*

Term 1	Term 2	Term 3
!computer &	!crime &	detection
!computer &	crime &	!detection
!computer &	crime &	detection
computer &	!crime &	!detection
computer &	!crime &	detection
computer &	crime &	!detection
computer &	crime &	detection

* The ''!'' indicates negation (topic 95, $K = 3$ terms).

such file is shown in Table 2. *mgquery* was run with options which make it easy to extract the number of items in each atom separately. In effect, *mgquery* is being used not to perform information retrieval, but to find the number of relevant (or not relevant) documents belonging to each Boolean atom of the collection.

Using the results of these queries, we can determine the size of each atom, the probability of relevance in each atom, and the probability of relevance of documents indexed by each keyword, for all the atoms combined. This information is provided to the MEP method and the estimated distribution given by the MEP is compared with observation.

5. Evaluation Criteria and Results

For each of the 50 queries considered, we break down the results according to the number of keywords used to describe the atoms. For example, if there are 5 keywords selected to represent the topic, we can have 2, 3, 4, or 5 keywords in a test of the MEP. In addition, for ''non-relevant'' documents, we tried three random samples from the TREC collection. These are referred to as the 0.1% random sample, the 1% random sample, and the 10% random sample. For each query, all relevant documents are combined in turn with each of these random samples which are considered as non-relevant documents. Using the combined documents and the keywords, the observed distribution of atoms and the parameters for the MEP are calculated. The distribution is also estimated by the MEP and compared. Table 3 shows an example of the observed distribution and the distribution estimated by the MEP.

5.1. Evaluating the Strong MEP

By the Strong MEP, we mean the hypothesis that the distribution of documents across atoms estimated by the MEP is equal to the distribution found by observation, i.e.,

H_0: The observed distribution is equal to the distribution given by the MEP.

Such a prediction can be tested by using the chi-square test for the agreement between the actual distribution and the distribution given by any model. In our case, the model is the MEP with constraints due to the distribution of relevant documents in sets described by a single term. Computational results, not reported here, suggest that the distribution is not well described by this model. In general, results are poorer for larger data sets, and poorer for combinations of five terms than of four, and poorer for combinations of four terms than of three, and so on. Of course, if the strong MEP were true, there would be no need for further tests, as the distribution determines all measures of performance.

5.2. Evaluating the Rank MEP

By the Rank MEP, we mean the hypothesis that the MEP will rank the Boolean atoms in the true decreasing order of the ratio of ''good'' document to ''bad'' documents. We test this using both the Spearman rank correlation (ρ) and a computation of average rank deviation.

To interpret the results of our calculations, we need some alternative models for ordering the Boolean atoms. We have considered two extremes. One, which might be called the ''divine alternative'' orders the atoms in strictly decreasing order of the odds ratio of relevance. No other arrangement can give a higher score, on any sensible test. The other, which might be called the ''naive ordering method'' uses the observed log odds ratio for each term. In particular, atoms are ranked in decreasing order of the sum of the probabilities corresponding to each of the terms that are present.

In a third alternative, called the ''lexicographic ordering method'' each term is assigned a ''probability of relevance score,'' such as V_A, V_B, etc. The atoms are then ranked lexicographically with respect to these scores.

TABLE 3. An observed distribution and the distribution estimated by the MEP.*

		Observed distribution		Estimated distribution by the MEP	
Boolean atom	Fraction of documents	Not relevant $p(i, 0)$	Relevant $p(i, 1)$	Not relevant $p(i, 0)$	Relevant $p(i, 1)$
$\bar{A}\bar{B}$	0.1470	—	—	—	—
$\bar{A}B$	0.4576	0.4479	0.0097	0.4337	0.0240
$A\bar{B}$	0.0886	0.0867	0.0019	0.0736	0.0150
AB	0.3067	0.2892	0.0175	0.3033	0.0034

* Topic 243, 0.1% r.s., total 1,027 documents, term A = government, term B = energy.

Note that, since we are not interested in the documents which have neither of two index terms A and B, the distribution in the Boolean atom $\bar{A}\bar{B}$ is not calculated.

TABLE 4. Rank correlation.

0.1% r.s.	Cases	Mean (*SD*) MEP	Naive	Lexicographic
Total	523	*.8991 (.1932)	.7614 (.2701)	.7347 (.3041)
NKEY 2	256	*.9468 (.1863)	.8154 (.3080)	.7788 (.3530)
NKEY 3	186	*.8748 (.1931)	.7411 (.2205)	.7219 (.2527)
NKEY 4	70	*.8022 (.1781)	.6494 (.1903)	.6369 (.2040)
NKEY 5	11	*.8173 (.1083)	.5582 (.1570)	.5470 (.1577)

1% r.s.	Cases	Mean (*SD*) MEP	Naive	Lexicographic
Total	523	.7757 (.3581)	*.7840 (.2670)	.7534 (.3019)
NKEY 2	256	*.8672 (.3926)	.8359 (.2949)	.7993 (.3438)
NKEY 3	186	.7153 (.3070)	*.7655 (.2296)	.7392 (.2590)
NKEY 4	70	.6211 (.2691)	*.6754 (.2038)	.6542 (.2091)
NKEY 5	11	*.6539 (.2390)	.5781 (.1932)	.5590 (.1794)

10% r.s.	Cases	Mean (*SD*) MEP	Naive	Lexicographic
Total	523	.6733 (.4180)	*.7857 (.2654)	.7521 (.3045)
NKEY 2	256	.8354 (.4023)	*.8433 (.2917)	.8008 (.3492)
NKEY 3	186	.5296 (.4021)	*.7599 (.2262)	.7316 (.2577)
NKEY 4	70	.4885 (.3024)	*.6754 (.2063)	.6568 (.2097)
NKEY 5	11	.5061 (.2069)	*.5830 (.1956)	.5694 (.1753)

* Represents the highest rank correlation in each row.

The naive and lexicographic methods are similar, but give different orders in a case such as:

$$V_A = .6 \quad V_B = .5 \quad V_C = .4 \quad V_D = .1 \qquad (7)$$

Lexicographically the Boolean atom of ''*A* and *D*'' precedes the atom of ''*B* and *C*.'' Under the sum rule, they are reversed.

The results are summarized in Table 4. The entries in Table 4 are the statistics for the values of the Spearman Rank correlation for all cases with a given number of keys, run on a given size sample. In each case, the computed order of the atoms is compared with the ideal order of the atoms, based on the observed distribution of relevant documents. For example, of the 70 cases with 4 key terms, on the 1% random sample, the average value of ρ is 0.6211 for the MEP, 0.6754 for the Naive method, and 0.6542 for the Lexicographic method. The standard deviations of each of these results are shown in parentheses following the means. The MEP method works well in the case of small samples and the Naive method behaves well for large samples.

The same array of results is shown scored by the average rank deviation in Table 5. In general, under this method of scoring, the MEP fares somewhat better than under the Spearman method. Note that in Table 5, small scores represent a good fit. While in Table 4, the MEP has the best average fit in 6 of the 12 rows of the table, in Table 5, the MEP has the best fit in 10 of the 12 cases. The rank correlation is an L_2 measure, which is more sensitive to outliers, which may account for this difference.

5.3. Evaluating the MEP by Expected Average Precision

Let R be a set of ''relevant'' documents given a query q, and G be the number of ''relevant'' documents. To evaluate the MEP using a standard measure of performance, we have selected the expected value of the so-called exact average precision. This is computed, for a ranked retrieved list L, in terms of the position occupied by each relevant document. A sublist is a consecutive set of documents from a list, in the same order in which they appear in the list. A standard sublist is one containing the first document of the list. Formally, for each document $d \in L$, we can define the smallest standard sublist containing that document, $L(d)$. The size of that sublist, $n(d)$, is the rank of item d in the list. Let $g(d)$ be the number of ''relevant'' documents in $L(d)$. In terms of this, the exact average precision is given by:

$$p_{ave} = \frac{1}{G} \sum_{d \in R} \frac{g(d)}{n(d)} . \qquad (8)$$

We need to estimate the contribution to p_{ave} resulting from the appearance of a sub-list with a known ratio of relevant

TABLE 5. Average rank deviation.

0.1% r.s.	Cases	Mean (*SD*) MEP	Naive	Lexicographic
Total	523	*.5271 (.8179)	.9936 (1.2919)	1.0161 (1.3232)
NKEY 2	256	*.0690 (.2067)	.2449 (.3531)	.2709 (.3900)
NKEY 3	186	*.5514 (.4692)	1.0207 (.5322)	1.0369 (.6071)
NKEY 4	70	*1.7000 (.8180)	2.7657 (.8478)	2.7904 (.9575)
NKEY 5	11	*3.3138 (1.2573)	6.6833 (1.3386)	6.7156 (1.5398)

1% r.s.	Cases	Mean (*SD*) MEP	Naive	Lexicographic
Total	523	*.8147 (1.2002)	.9031 (1.2653)	.9525 (1.3042)
NKEY 2	256	*.1328 (.3438)	.2162 (.3433)	.2422 (.3832)
NKEY 3	186	*.8894 (.6838)	.9047 (.5891)	.9577 (.6414)
NKEY 4	70	*2.4629 (1.1053)	2.5562 (.9974)	2.6591 (1.0016)
NKEY 5	11	*4.9326 (2.3289)	6.3403 (1.8837)	6.5337 (1.7148)

10% r.s.	Cases	Mean (*SD*) MEP	Naive	Lexicographic
Total	523	1.0750 (1.4487)	*.9013 (1.2634)	.9526 (1.2982)
NKEY 2	256	*.1745 (.3745)	.2084 (.3405)	.2383 (.3873)
NKEY 3	186	1.2581 (.7940)	*.9201 (.5861)	.9762 (.6418)
NKEY 4	70	3.0724 (1.3390)	*2.5410 (1.0402)	2.6401 (1.0445)
NKEY 5	11	*6.2257 (1.8568)	6.2726 (1.9191)	6.4398 (1.6963)

* Represents the smallest rank deviation in each row.

to not relevant documents. This problem has been addressed by Cleverdon (1970) a few years ago. A more precise solution is derived below.

Suppose there are K index terms in a test, and therefore there are 2^K Boolean atoms. We refer to a particular ordering of the atoms, for example by decreasing order of the ''relevance'' probability of each atom as predicted by the MEP, as a complete list or ''blocking.'' A ''blocking'' will, in general, be composed of a succession of blocks, each of which contains some number of documents, of which some (weakly smaller) number are relevant.

Assume that the i^{th}, $i = 1, 2, \ldots, 2^K$, block contains n_i documents, of which g_i are relevant. When there are g_i relevant documents distributed among n_i documents in the i^{th} block, they may be arranged in $\binom{n_i}{g_i}$ ways. The position occupied by the k^{th} relevant document is precisely the k^{th} order statistic $u^{(k)}$ of the corresponding arrangement, which is an element of the space of all possible arrangements (see, for example, David, 1970). The sum over all relevant documents, and over all possible combinations of placements of those documents contributes to p_{ave} as follows.

Let the number of relevant documents which have already appeared before the i^{th} block be G_i, and let the total number of documents which have already appeared before the i^{th} block be N_i, i.e.,

$$G_1 = 0, \quad N_1 = 0$$
$$G_i = \sum_{j<i} g_j, \quad N_i = \sum_{j<i} n_j, \quad i = 2, 3, \ldots, 2^K. \quad (9)$$

The contribution of a k^{th} relevant document of the i^{th} block, appearing in position $u^{(k)}$, to the sum defining p_{ave} is given by:

$$\frac{G_i + k}{N_i + u^{(k)}}. \quad (10)$$

This equation represents the fact that when the k^{th} relevant document is encountered, the total number of relevant documents at that point is $G_i + k$, while the total number of documents encountered (the position of this k^{th} document in the list) is $N_i + u^{(k)}$. That is, the position of this k^{th} document in its block is precisely the corresponding order statistic.

The sum over all placements of the relevant documents in the i^{th} block may be written as:

$$\sum_{combinations} = \sum_{k=1}^{g_i} \sum_{u^{(k)}} \sum_{Number\ of\ cases\ N(k,u^{(k)})} . \quad (11)$$

In general, $k \le u^{(k)} \le n_i - (g_i - k)$. The number of cases $N(k, u^{(k)})$ is precisely the number of ways that exactly $k - 1$ relevant documents can be placed in $u^{(k)} - 1$ slots, and at the same time, $g_i - k$ relevant documents can be placed in $n_i - u^{(k)}$ slots.

$$N(k, u^{(k)}) = \binom{u^{(k)} - 1}{k - 1}\binom{n_i - u^{(k)}}{g_i - k}. \quad (12)$$

Assembling all of this, we find the expected average precision of the i^{th} block, $p_{ave}(g_i, n_i; G_i, N_i)$, which is a contribution to p_{ave}. This is the contribution given by each possible arrangement, divided by the total number of arrangements, and averaged over the g_i relevant documents.

$$p_{ave}(g_i, n_i; G_i, N_i) = \frac{1}{g_i} \sum_{k=1}^{g_i} \sum_{u^{(k)}=k}^{n_i-(g_i-k)} \frac{\binom{u^{(k)}-1}{k-1}\binom{n_i-u^{(k)}}{g_i-k}}{\binom{n_i}{g_i}} \times \frac{G_i + k}{N_i + u^{(k)}}. \quad (13)$$

Finally, the expected value of p_{ave} of a blocking can be computed as follows:

$$p_{ave}(blocking) = \frac{1}{G} \sum_{i=1}^{2^K} g_i \, p_{ave}(g_i, n_i; G_i, N_i). \quad (14)$$

Substituting Equation 13 into Equation 14 yields a complete expression for the expected value (averaging over all the ways that the relevant documents could be distributed within their blocks) of the measure p_{ave}. This measure is the average, over all the relevant documents, of the precision *computed at the point where each relevant document appears.*

The calculations were done using Stirling's approximation, and suppressing all configurations in which the k^{th} relevant document would be at a position in the upper 1×10^{-9} tail of the distribution of the k^{th} order statistic. For the 10% random sample (with more than 10,000 documents after excluding the atom which has neither of index terms), the computation took less than 3 hours on an Intel 50 MH 486 DX computer.

The results of these computations are summarized in Table 6. The entry in the table is normalized by comparing it to the expected average precision of the optimum ordering of the atoms and the expected average precision of ignoring the key terms and choosing documents at random. The normalized score is called the ''efficiency'' of the ordering of the atoms, η.

$$\eta = 100 \frac{\{p_{ave}(blocking) - p_{ave}(random\ blocking)\}}{\{p_{ave}(optimum\ blocking) - p_{ave}(random\ blocking)\}}. \quad (15)$$

For example, in the 245 cases of atoms defined by two terms, in the 0.1% random sample, the mean over all cases of the efficiency η for the MEP method was 99.14, slightly exceeding the corresponding average for the other two methods.

Using the expected average precision, we find that the MEP does best in 5 of the 12 configurations examined. Once again, for the largest sample, the MEP does not fare well.

6. Conclusions

We find that, using an extensive collection such as the TREC, it is possible to do a thorough test of the Maximum Entropy Principle, as it can be applied to IR. The results show that the MEP performs better than two simpler methods (the Naive and Lexicographic methods) for small collections. It does well on larger collections only when a particular choice of the scoring method is chosen: Average absolute rank deviation of the ordering of atoms.

From the point of view of retrieval, it would be sufficient to get the Boolean atoms in the correct rank order. The MEP seems promising in this regard, getting nearer to the ideal order at least as often as the two comparison methods.

However, simulating the results of a computation of the average precision score leads to a somewhat less positive result, as MEP is best in only 5 of the 12 blocks of the study design, and never for the largest set.

In general, the MEP does best for small collections and small numbers of terms, at least in the examples studied here. On the other hand, we may look for reasons why the present experiment is not a true test of the MEP. The most serious limitation is that we know the relevance judgments only for a set of documents which have been presented in TREC conferences. Insofar as the systems used all work on a limited set of principles, the documents studied here (at least the ones for which evaluation are known) are not at all a random sample of the documents in the database. Yet such a random sample is what ought to be used to test assumptions about distributions.

In addition, the MEP really specifies that the distribution of relevant documents maximizes entropy subject to some conditions. Following Cooper, we have used only conditions referring to (1) the number of documents in each Boolean atom, and (2) the fraction of documents which are relevant, in a subset defined by the presence of a single term. In principle, one might include information on the number of relevant documents occurring in sets defined by co-occurrence of pairs of terms. We have chosen not to.

In spite of these possible caveats, it is our judgment that the Maximum Entropy Principle does not describe the distribution of terms in relevant (respectively nonrelevant) documents with an accuracy sufficient for it to be used in information retrieval systems.

In particular, we feel that the data show a clear discouraging trend. The real case of interest is the case of very large data sets. Our data clearly show that the MEP works best for small sets, and seems to be progressively worse for larger sets. The prospect for having it turn out to be effective for enormous sets is therefore slim or non-existent.

Our analysis has applied the MEP to a binary classification, based on the presence or absence of terms in a text. All effective modern systems make use of much more information, as they use the frequency with which terms appear in a text, weighted and transformed in various ways, in an estimate of relevance. It might be imag-

TABLE 6. Efficiency of the expected average precision..

		Mean (*SD*)		
0.1% r.s.	Cases[a]	MEP	Naive	Lexicographic
Total	510	*98.43 (4.81)	96.17 (11.02)	96.23 (11.01)
NKEY 2	245	*99.14 (4.18)	96.29 (13.72)	96.29 (13.72)
NKEY 3	184	*98.01 (5.71)	96.30 (8.36)	96.31 (8.37)
NKEY 4	70	*97.14 (4.22)	95.51 (6.52)	95.81 (6.47)
NKEY 5	11	*97.62 (1.76)	95.66 (2.74)	96.21 (2.84)
		Mean (*SD*)		
1% r.s.	Cases[b]	MEP	Naive	Lexicographic
Total	510	89.75 (23.15)	*93.77 (18.43)	92.25 (32.16)
NKEY 2	245	*94.05 (25.40)	93.99 (24.72)	91.41 (45.31)
NKEY 3	184	88.10 (20.08)	*94.26 (10.31)	93.87 (10.44)
NKEY 4	70	81.11 (19.03)	*92.25 (7.21)	91.43 (7.83)
NKEY 5	11	76.76 (21.10)	*90.05 (6.10)	89.00 (6.23)
		Mean (*SD*)		
10% r.s.	Cases[c]	MEP	Naive	Lexicographic
Total	497	82.71 (27.32)	*91.49 (19.71)	89.80 (22.76)
NKEY 2	233	93.73 (22.86)	*95.13 (22.31)	93.12 (27.07)
NKEY 3	183	76.84 (27.21)	*90.65 (16.29)	89.17 (17.85)
NKEY 4	70	65.76 (25.65)	*83.80 (15.44)	82.59 (16.14)
NKEY 5	11	55.08 (24.15)	*77.24 (16.80)	75.75 (16.06)

* Represents the highest expected average precision in each row.
[a] In 13 cases, $p_t = p_o$ and the efficiency η could not be computed.
[b] In 13 cases, $p_t = p_o$ and the efficiency η could not be computed.
[c] In 26 cases, $p_t = p_o$ and the efficiency η could not be computed.

ined that using this more refined representation, together with the MEP, would be more effective than what we have found. We have chosen not to explore this possibility.

Let us recall that the MEP moves beyond stochastic term independence (which is the basis for the Robertson Sparck-Jones weighting scheme) only to the extent required by the observed distribution of the sizes of the atoms. The underlying assumption of independence is, itself, quite suspect. In fact, terms are clustered together, by authors, in order to express meaning. Thus, in spite of the appeal which MEP holds from a computational point of view, it may simply not be reflective of the real logical structure which underlies the difference between relevant and non-relevant texts. In conclusion, we believe that the evidence cannot support the strong claim that the MEP accurately describes the distribution of terms across relevant and non-relevant texts. Nor does it support the weaker claim, that computing according to the MEP will lead to enhanced information retrieval.

7. Discussion

Selection of terms was not scientifically investigated in this work. Reasoning that if the MEP is ''true,'' it will be true for every choice of terms, we simply selected terms that seemed indicative of the problem described in each TREC topic. For completeness, the list of terms in each topic is included as an Appendix.

While these results are somewhat disappointing, they open some interesting possibilities. We have found that the MEP is computationally tractable for collections of 100,000 documents. Most of the computational effort went into determining the ''ground truth'' about the distribution of relevant documents. In a practical application, that would be estimated from a sample no larger than the ones used here.

The important question is: Does the MEP, in general, give us anything that could not be obtained from the Naive method. The present answer seems to be ''no.''

8. Acknowledgments

This research was completed during the visit of J. J. Lee to Alexandria Project Laboratory (APLab) in the School of Communication, Information and Library Studies (SCILS) at the State University of New Jersey, Rutgers, under partial support of the Korea Research Foundation. He thanks the APLab and SCILS for providing an excellent research environment and for their hospitality.

We thank Kwong Bor Ng for providing very substantial assistance in the management of the TREC collection, and maintaining the mgquery software.

9. Appendix: List of Terms in Each Topic

Topic no.	Terms				
001	antitrust	pending	violation	investigation	
003	new	joint	venture	Japanese	
004	debt	rescheduling	developing	countries	
005	dumping	charges	Japan		
006	debt	relief	developing	country	
011	space	project	goals		
012	water	pollution	risk	source	
023	legal	remedies	misuse	accidental	agrochemical
024	new	medical	technology	treatment	drugs
044	staff	reduction	computers	communication	
053	leveraged	buyout			
054	satellite	launch	contracts		
058	predict	rail	strike	ongoing	
068	hazards	safety	fine-diameter	fiber	
077	poaching	method	wildlife		
078	Greenpeace	activity			
082	genetic	engineering	application	attitude	
094	computer	crime			
095	computer	crime	detection		
100	transfer	regulate	high-tech	non-communist	
108	Japanese	protection	domestic	market	
111	control	nuclear	proliferation		
114	non-commercial	satellite	launch		
118	international	terrorist	activities		
119	activities	against	international	terrorists	government
121	death	cancer	U.S.		
123	control	environment	chemicals	carcinogenic	
125	anti-smoking	actions	government		
126	ethics	medical	technology		
142	government	grain	policies	international	relation
154	oil	spills			
161	acid	rain			
173	smoking	bans			
185	reform	welfare	system	U.S.	
187	demise	purchase	publisher		
189	motives	murder			
192	oil	spill	cleaning		
193	dangers	toys	safety	activities	
194	money	earned	writers		
195	stock	market	computer	trading	perturbation
202	nuclear	proliferation	treaties	violation	monitoring
207	Quebec	independence			
211	driving	intoxicated	regulation	effective	
221	church	government	community	youth	drug
222	capital	punishment	crime		
224	lower	high	blood	pressure	effects
228	environment	recovery	pollution		
237	energy	automobile	additive	decrease	pollution
240	control	agreement	technology	equipment	terrorism
243	government	fossil	energy	corporation	utility

10. References

Cleverdon, C. (1970). Aslib proceedings 1967. In T. Saracevic (Ed.), *Introduction to information science*. New York: R. R. Bowker Company (pp. 608–620).

Cooper, W. S. (1983). Exploiting the maximum entropy principle to increase retrieval effectiveness. *Journal of the American Society for Information Science, 34,* 31–39.

Cooper, W. S., & Huizinga, P. (1982). The maximum entropy principle and its application to the design of probabilistic retrieval systems. *Information Technology: Research and Development, 1,* 99–112.

David, H. A. (1970). Order statistics. New York: Wiley.

Harman, D. (Ed.). (1996). *Proceedings of the 4th Annual Text Retrieval Conference.* Gaithersburg, MD: NIST. (NIST SP 500-236)

Harman, D. (Ed.). (in press). *Proceedings of the 5th Annual Text Retrieval Conference.* Gaithersburg, MD: NIST. http://www-nlpir.nist.gov/TREL/t5_proceedings.html

Kantor, P. B. (1984). Maximum entropy and the optimal design of automated information retrieval systems. *Information Technology: Research and Development, 3,* 88–94.

Kantor, P. B. (1994). Information retrieval techniques. In M. Williams (Ed.), *Annual review of information science and technology* (pp. 53–90). Medford, NJ: Learned Information.

Kantor, P. B., & Lee, J. J. (1986). The maximum entropy principle in information retrieval. In F. Rabitti (Ed.), *The Proceedings of the 1986 ACM Conference on Research and Development in Information Retrieval,* Pisa, Italy (pp. 269–274).

Lasdon, L. S., & Waren, A. D. (1979). Generalized reduced gradient software for linearly and nonlinearly constrained problems. In H. Greenberg (Ed.), *Design and implementation of optimization software.* Sijthoff and Noordhoff.

Lee, J. J., & Kantor, P. B. (1991). A study of probabilistic information retrieval systems in the case of inconsistent expert judgments. *Journal of the American Society for Information Science, 42,* 166–172.

Lewis, D. D. (1996). The TREC4 Filtering Track. In D. Harman (Ed.), *Proceedings of the 4th Annual Text Retrieval Conference* (pp. 165–180). Gaithersburg, MD: NIST. (NIST SP 500-236).

Ryan-McFarland Corp. (1986). *RM/FORTRAN user's guide* (Version 2.10). Rolling Hills Estates, CA.

Witten, I. H., Moffat, A., & Bell, T. C. (1994). *Managing gigabytes.* New York: Van Nostrand Reinhold.