Application of Logical Analysis of Data to the TREC6 Routing Task

Endre Boros
RUTCOR, Rutgers University
boros@rutcor.rutgers.edu

Paul B. Kantor
Jung J. Lee*
Kwong Bor Ng
Di Zhao
Alexandria Project Lab, SCILS, Rutgers University
{kantor, jungjlee, kbng, dizhao}@scils.rutgers.edu

1. The Logical Analysis Approach in the Official Runs

Our approach to TREC6 has explored the possibility of building complex Boolean expressions which represent the classificatory information present in the training data. The positive (i.e. judged relevant), and negative (i.e. judged not relevant) documents are studied separately, using Church's measure of "non-Poissonicity" (Church & Gale, 1995) to identify promising terms for classification.

In the official runs, statistics are produced using the ***MG*** (Witten, Moffat, Bell, 1994)) search engine, and the terms are in fact stems, rather than complete terms. The top 25 terms selected from the positive and negative examples are merged, to form a list with no more than 50 terms. The ***MG*** retrieval system is used (massively) to transform every judged document into a Boolean vector with one component for each distinct classification term. The RUTCOR ***LAD*** program (Boros, Hammer, Ibaraki, Kogan, Mayoraz, & Muchnik, 1996) is used (twice for each topic), with several modifications, to search exhaustively for Boolean prime implicants which characterize the positive and the negative examples. Due to computer speed limitations, we have limited the search in our official submissions to terms of order three (i.e terms such as ABC', where C' denotes the absence of term C). Each pattern which matches some positive (respectively, negative) examples is given a weight determined by the number of examples that it matches.

2 Detailed Procedures of the Official Runs

2.1 Training

For each topic, we used ***MG*** to index all the judged relevant documents to build a index structure, and to compute the term frequencies and document frequencies of all word-stems. We

* Permanent address: Department of Statistics, Soong Sil University, Seoul, Korea.

selected 25 word-stems according to the Church criterion (Church & Gale, 1995) on distributions of term frequencies and document frequencies. We did the same for the judged non-relevant documents. For topics with more than 50 Mbytes of judged non relevant documents we randomly selected 50% of the judged documents for MG to index. (Topics: 77, 78, 82, 94, 95, 100, 108, 118, 119, 123, 125, 126, 128, 142, 161, 173, 187, 194, 228, 240, 282). This yields 25 word-stems from relevant documents and 25 word-stems from non relevant documents. Each stem was submitted as a Boolean query, using ***MG***. This produced a list of documents in which the term appeared. These lists are next combined to form a single file in which each relevant document is represented by a single row of 0s and 1s, where 1 signifies that the stem labeling the corresponding column appears at least once in the document. This is the form of case representation accepted by ***LAD***. We do the same for the non-relevant documents, producing a second array of cases.

For each topic, we concatenate the files for the relevant and no-relevant training examples, the degree is set to ***k***, and ***LAD*** finds all Boolean monomials with ***k*** literals, matching some relevant document vectors and no non-relevant document vectors. These are the positive patterns of the topic. Negative patterns are defined correspondingly to match non-relevant documents. Thus ***LAD*** provides the foundation for Boolean classification rules.

The process takes time exponential in ***k.*** We were limited to ***k*** = 3 by time constraints. For topic 44, we could not find any positive patterns. We used the patterns file to assign a weight for each pattern, equal to the number of training documents that fit the pattern. Note that due to limitations of person-power and time, our training phase did not contain any evaluation and tuning of the numerous parameters in both attribute selection and ***LAD*** pattern-finding. More details about implementation and algorithms are in section 4.

2.2 Testing

We used ***MG*** to index the test collection of routing documents. We used a stepwise fusion process to produce, from the Boolean patterns, ranked lists. The positive patterns were used to produce a reduced set (except for topic 44). The reduced set is the union of all documents retrieved by any of the patterns. We used two methods to fuse the documents into one single ranked list. The first method is a "quorum" method. Documents are ranked in decreasing order of the number of patterns which retrieved them. The second is weighted fusion. Each pattern has a weight equal to the number of training documents that it covers. The score assigned to a document is the sum of the weights of all the patterns which cover it. Both quorum and weighted scores were also computed for each document, using the set of negative patterns.

Our submission was the top 1000 documents of the positive rank list. We planned to eliminate or re-order those documents retrieved by positive queries according to the ranked list produced by negative pattern queries until the positive document ranked list had only 1000 documents, i.e., we would eliminate from the positive document list the documents that are also in the negative document list and eliminate those with the highest rank of the negative document list first, and so on until the positive document list contains only 1000 document. We found that using this method to eliminate documents could easily eliminate much more than desired number of documents. That is, at certain ranks in the negative list, we would acquire a batch of "knockouts" which brought the remaining list below 1000.

3 Results of the Official Runs

The results are not distinguished. Using the exact averaged precisions our results are occasionally worse than the "worst". We therefore concentrate on the precision at 100 documents. The weighted method performs better than the quorum method in 11 cases, and worse in 8. They are tied in 28. More importantly, the Quorum method produced the worst recorded result in 30 cases, and the weighted method did so in 29 cases. It is clear that the combination of decisions that we have made does not solve the routing problem. We suspect that several factors combine to produce this discouraging result.

4. The LAD Approach in the Non-Official Runs

After we submitted our official runs, we continued our experiments . We have implemented a method based on the Logical Analysis of Data, as it is described in Boros, Hammer, Ibaraki, Kogan, Mayoraz, & Muchnik, 1996 (see for further details, Boros, Hammer, Ibaraki & Kogan, 1997, Boros, Ibaraki & Makino, 1997, and, Boros, Ibaraki & Makino, 1998), with several modifications.

In these experiments, the algorithm we implemented consists of 4 phases. The first phase is a more or less standard indexing of the documents. We have used the ***SMART*** system (version 11.0, implemented on Sun Ultra-1, Solaris 2.5.1), and as a result we have obtained an indexed representation of the documents. Let us denote documents by d, and terms by t, and let us denote by $f(t,d)$ the number of occurrences of term t in document d. The length of a document d is

$$l(d) = \Sigma_t f(t,d)$$

and the relative frequency of a term t in a document d is

$$r(t,d) = \frac{f(t,d)}{l(d)}$$

We have indexed, for all 47 TREC-6 topics, most of the training documents. We exclude IRList digests, Usenet news groups documents, and Virtual World documents. These three collections contain relatively few relevant documents and they are not included in the Tipster document CD collection.

The second phase is a projection, in which we map the high dimensional frequency vector representation into a low dimensional binary representation, essentially following the ideas described in (Boros, Hammer, Ibaraki & Kogan, 1997), with some very important modifications. For a term t and a real number z, let us introduce a propositional statement of the form $X(d)$= "term t occurs with relative frequency higher than z in document d". Such a statement assumes a logical value (true or false, i.e. 1 or 0) for every document. By choosing such pairs (t_i,z_i) appropriately, and denoting the corresponding propositional variables by X_i for $i = 1,...,k$, we can map every document d into a binary vector $X^d = (X_1(d), ..., X_k(d))$. Ideally, one would like to select pairs (t_i,z_i) such that they "represent well" the particular topic, and one would think, those stating the particular topic are the best to choose. In our algorithm we instead use an automatic learning method for selecting such pairs, and **we did not use the topic descriptions**. For each potential pair (t,z) we computed two parameters:

$$R(t,z) = |\{ d \in RelevantTraining(Topic) : r(t,d) > z + GAP \}|$$

and

$$I(t,z) = |\{ d \in NonrelevantTraining(Topic) : r(t,d) < z - GAP \}|$$

where GAP is a preselected small positive constant. Finally we set $S(t,z) = R(t,z)*I(t,z)$, i.e. $S(t,z)$ counts the pairs of (relevant,non-relevant) documents in the training set of the considered topic, which are "properly" distinguished by the logical statement corresponding to the pair (t,z) with a separation GAP.

In the second phase of the algorithm, we select the smallest set I, indexing pairs (t_i,z_i), for which the separation value $S(t_i,z_i)$ is high and such that for every pair consisting of a relevant document d and a non-relevant document d' in the training set the condition

$$(r(t_i,d) > z_i + GAP) \ AND \ (r(t_i,d') < z_i - GAP)$$

is satisfied by at least M different indices i ($i \in I$), where M is an input parameter. Typically M = 10, or so. In other words, we would like to have a binary encoding of the documents, which is as short as possible, and such that the vectors X^d and $X^{d'}$ are very different, whenever the relevance states of the documents d and d' are different. Since this optimization problem is difficult to solve, we implemented an efficient (polynomial time) approximation to solve it. In our experiments we used GAP = 0, M = 10, and to decrease the chances of overfitting, we have used only a randomly selected subset (50-80%) of the training documents. The values we obtained for k varied between 30 and 150 for the different topics. Let us add that for most topics just by reading the terms t_i, $i = 1,...,k$, one could get a very good idea of what the topic was about!

In the third phase, we were looking for simple logical rules, in terms of the binary variables X_i, $i = 1,...,k$, characterizing relevance (or non-relevance) well. For instance, if term t_1 is "Japan" and term t_2 is "dump", then X_1 AND X_2 is the simple statement of "the relative frequency of *Japan* is more than z_1 AND the relative frequency of *dump* is more than z_2" and the truth of this for a document d may be a good indicator that d is about "dumping by Japanese companies". More precisely, let us call an elementary conjunction $P = X_{i1}$ *AND* X_{i2} *AND ...* (in which some variables may appear with a negation) a pattern, if $P(d)$ = 0 for all non-relevant documents $d \in$ ***Nonrelevant(Topic)*** in the training set, and $P(d)$ = 1 for some relevant documents. We say that "the pattern P is triggered" for document d if $P(d)$ = 1. In other words, a pattern is a statement which confirms the relevance of some documents, while not raising any false alarms on the non-relevant documents. Let us denote by $C(P)$, called the coverage of P, the number of relevant documents in the training set for which P = 1. Obviously, the higher $C(P)$ the more we can trust P (that is, the greater its recall) , and the more we find its existence surprising!

In this phase of our method, we first generate a pattern ***P*** for every relevant document ***d*** such that ***P(d)*** = 1, and the coverage ***C(P)*** is as high as possible. Since this is again a very hard optimization problem, we employ a fast (polynomial time) approximation algorithm. Let us call the patterns obtained in this stage positive patterns. (Of course, we filter out patterns dominated by others. A pattern is "dominated" if it contains a subpattern which is as effective as it is.) We then interchange the role of relevant and non-relevant documents, and generate a "negative" pattern for every non-relevant document, analogously, i.e., ***N*** is a negative pattern if ***N(d)*** = 1 only for (some) non-relevant documents; and ***P*** is a positive pattern if ***P(d)*** = 1 only for (some) relevant documents.

Finally, we select a smallest subset of these positive and negative patterns such that for every document ***d*** in the training set at least *N* of these selected patterns are triggered. (These triggered patterns will be only positive patterns if ***d*** is relevant, and must all be negative patterns if ***d*** is non-relevant.) In our experiments we choose $N = 5$.

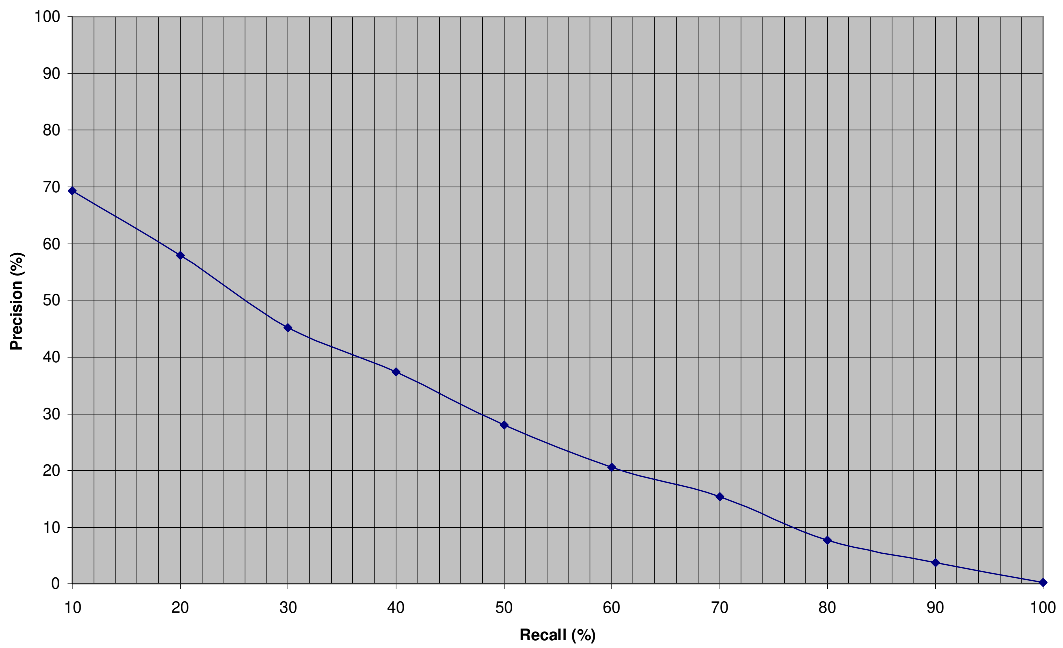
In the last phase we compute a score for every document in the test set by setting

$$S(d) = \sum_{P\ positive} P(d) - \sum_{N\ negative} N(d).$$

Intuitively, if the training set represent the given topic well, this score must be positive for relevant documents, and negative for non-relevant documents.

In our experiments the average precision over all 47 topics was about 28%, which is about the same as the median of all other methods at the TREC-6 meeting. Out of the 47 topics our average precision was better than the median 27 times, and was below the median 20 times.

Average over 47 topics



5. Discussion

This effort represents a first step towards automating the routing process in a way which reflects the natural human preference for (and documented effectiveness of) Boolean formulations as a basis for ad hoc retrieval. There are many ways in which the present first attempt can be expanded, including the search for effective synonyms appearing in the Boolean implicants, and more subtle methods for combining the patterns found. As an example, since all the judged documents in the TREC setting have been retrieved by other systems in prior years, one might have an "Inclusion rule" built on all of the judged documents (contrasted to a random selection of unjudged documents), followed by an "Exclusion step" based on patterns which resolve the judged relevant from the judged non-relevant. Conceptually, this approach is based on the belief that Boolean combinations of "terms" which are in turn surrogates for "concepts" are a powerful representation of texts when the goal is to estimate relevance.

While it is customary to treat the routing task as deriving most of its information from the judged documents, we intend to examine this assumption, in our setting, by looking next at the terms of the topic, to see whether there are any potential useful terms that were not discovered by our process. If there seem to be any such terms we will test what happens when they are added to the set of basic variables. Since many TREC systems are vector based, we conjecture that this effect is most likely to occur if a Topic specified that the document is to be "not about such and such".

Many choices were made in the press of time, and without any systematic evaluation of the alternatives. It is our present belief that the following factors may explain the improvement between our official runs and the subsequent experiments reported here.

1. Originally the stems forming the basis for binarization of the data were chosen on a distributional criterion. Now they are chosen on the basis of power in separating the positive and negative instances in the training set.

2. Our choice of degree 3 was due to resource limitations. The current method finds some patterns of very high degree.

3. Our training set consists of documents which were retrieved in prior years, by systems which behave in roughly similar fashions. Thus our training procedure may not be the most logical one. An alternative is a two-step procedure: (1) find patterns which distinguish retrieved documents from all documents; (2) find patterns which distinguish the non-relevant retrieved documents from the relevant ones. This alternative procedure corresponds more faithfully to the way in which the patterns were used in our official submission, and it seem reasonable that training towards this purpose will produce better results.

Formally, the ***LAD*** method, as opposed to vector classifiers, or even quadratic classifiers, supports retrieval of substantially distinct clusters of relevant documents, in the underlying vector space with word stems for axes. This is, in principle, attractive, as it exploits a special feature of normal human searching. However, the present results show that the methods for finding the clusters our patterns must be made substantially more powerful to be competitive with todays state-of-the-art vector based retrieval systems.

Acknowledgements : This work has benefited enormously from conversations with Peter L. Hammer, Alex Kogan, Slava Brover, and Ken Church.

References

Boros, E, Hammer, P. L., Ibaraki, T., Kogan, A., Mayoraz, E. and Muchnik, I. (1996). An Implementation of Logical Analysis of Data. Rutcor Research Report 22-96 RUTCOR, Rutgers University.

Boros, E, Hammer, P.L., Ibaraki, T., & Kogan, A. (1997).Logical analysis of numerical data. Mathematical Programming, 79, 163-190.

Boros, E., T. Ibaraki, T, & Makino, K.(1998) Error-free and best-fit extensions of partially defined Boolean functions.Information and Computation, 140 (2), 254-283.

Boros, E., Ibaraki, T., & Makino, K. (1997). Monotone extensions of Boolean data sets, in: Algorithmic Learning Theory -- ALT'97 (M. Li and A. Maruoka, eds.).Lecture Notes in Artificial Intelligence 1316, Springer, pp.161-175.

Church, K. F. & Gale, W. A. (1995). Inverse Document Frequency (IDF): A measure of deviation from Poisson. Proceedings of the Third Workshop on Very Large Corpora.

Witten, I. H, Moffat A, Bell TC. (1994). Managing Gigabytes. New York: van Nostrand Reinhold.