Feature Selection and Batch Filtering

Endre Boros* and David J. Neu

May 16, 2003

Abstract

This year at TREC 2002 we participated in the batch filtering sub-task of the filtering track and investigated the use of rank-based feature selection techniques in conjunction with a very simple classification rule.

1 Introduction

The *Text REtrieval Conference* (TREC) is a yearly meeting in which various information retrieval systems are compared in several categories and in a semi-competitive environment (see e.g., [13]).

In the year 2002 filtering track a test collection of about 800,000 news documents were provided for evaluation together with 100 different topics, each consisting of a short description of some information need. In the batch filtering sub-task a much smaller training collection was also provided, including documents for each of the topics labelled as relevant or irrelevant. Systems were supposed to develop a static classifier for each of the given topics, based on the topic description and the labelled training documents, which then were applied to the larger test collection. Each of these classifiers read the test documents one-by-one, and judged them to be relevant or irrelevant for the considered topic. Those judged to be relevant were considered "retrieved", and were submitted to TREC for evaluation (see [13, 2] for more detail on the competition, on the rules of evaluation, and on the obtained results).

Let us emphasize first of all that efforts to attack the above classification problem are often complicated by the characteristics of textual data. Textual data is generally represented by using the terms in the text as features. Such data is inherently *high dimensional* — the number of features being potentially equal to the number of words in the English language. In addition, misspellings, the improper, or colloquial use of words, and the fact that many very common words (e.g. "a", "and", "the", etc.) are virtually useless for recognizing relevance, regardless of the information need, lead textual data to be *noisy*. Finally, the amount of information available to learn from is typically very small. For instance, out of the 100 topics only 2 had more than 100 relevant documents in the training collection of about 20,000 documents. In fact for more than half of the topics there were less than 10 relevant documents to learn from.

The aforementioned characteristics of textual data indicate that it might be possible to represent a document collection using only a subset of the original feature set which is much smaller than the original feature set, yet possesses properties which serve to facilitate the process of distinguishing relevant documents from irrelevant documents. Let us add that ideally, we would like to select the "most relevant" terms for a particular topic. However, "relevance" is a highly elusive notion, on which even human readers disagree frequently. There are several useful measures of relevance of attributes in the literature, primarily applied in game theory, for completely defined logical systems (see e.g., [8, 7, 12, 14]). Since feature selection became very important in contemporary ma-

*The work of the first author was partially supported by the Office of Naval Research (grant N00014-92-J1375) and by the KD-D group for a project at DIMACS on Monitoring Message Streams, funded through National Science Foundation grant EIA-0087022 to Rutgers University.

chine learning techniques (see e.g., [1, 4, 9, 10, 11]), many attempts were made to generalize such measures for the incompletely defined systems arising in machine learning tasks (see e.g., [3, 9, 5]). However, most of these attempts lead to questionable results. For instance [9] analyzed several possible measures and showed that in incompletely defined systems all these may lead to paradoxical and inconsistent conclusions.

Let us add that most machine learning algorithms employ explicitly (or implicitly) feature selection techniques (e.g., most document retrieval systems assign weights to the terms, eliminating in effect the influence of terms with small weights). Most of these feature selection approaches are based in fact not on the direct "relevance" of an attribute, but rather on the degree to which the attribute helps in distinguishing records in one class from those in others (see e.g., [1, 9, 11, 5]).

Our aim in this study was to develop a computationally very efficient, simple method, which is highly scalable for even much larger collections, and in which the developed classifiers are based only on a small number of selected terms. The idea we pursued was to employ a few simple heuristic measures of "distinguishing power" of individual terms for the generation of such small feature subsets, and then to train an extremely simple classifier on the training set represented only in terms of the selected small feature set.

2 Model

Assume that there are $n > 0$ distinct terms in the document collection and associate an index in $V = \{1, 2, \ldots, n\}$ with each of these terms. Letting $\mathbb{B} = \{0, 1\}$, we represent each document in the collection as an n-dimensional Boolean vector $\mathbf{x} \in \mathbb{B}^V$. Each component of $\mathbf{x}$ corresponds to one of the distinct terms in the document collection, with $x_i = 1$ if the i^{th} term is present in the document and $x_i = 0$ if the i^{th} term is absent from the document.

For a subset $S \subseteq V$, and vector $\mathbf{x} \in \mathbb{B}^V$, we shall let $\mathbf{x}[S] \in \mathbb{B}^S$ denote the projection of $\mathbf{x}$ onto S and for $X \subseteq \mathbb{B}^V$ we shall write $X[S]$ as the projection of X on S, that is, $X[S] = \{\mathbf{x}[S] \mid \mathbf{x} \in X\}$. For a subset $S \subseteq V$ let us denote by $\chi^S \in \mathbb{B}^n$ its *characteristic vector*, i.e. for which $\chi^S_j = 1$ iff $j \in S$.

For any given topic, we shall divide the training collection into the sets of relevant and irrelevant documents, denoted respectively by T and F. We shall assume that $T \cap F = \varnothing$, that is, there do not exist vectors $\mathbf{x} \in T$ and $\mathbf{y} \in F$ such that $\mathbf{x} = \mathbf{y}$.

A set $S \subseteq V$ is said to be a *support set* for T and F if it has the property that $T[S] \cap F[S] = \varnothing$. That is, S is a support set if each relevant document represented in terms of the selected features subset can be distinguished from each irrelevant document represented in terms of the selected features subset.

The document model described above does not preserve information about the order in which terms appear in the document and therefore is often referred to as the *bag-of-words* representation. In addition, the Boolean nature of this representation is a further simplification of the popular representation known in the information retrieval literature as the *vector space model*, in which the components x_i correspond to the (relative) frequency of the term in the document.

3 Measure of Separation

For a subset $S \subseteq V$, we measure the distance between the projections $T[S]$ and $F[S]$ of the sets T and $F \in \mathbb{B}^V$ onto $\mathbb{B}^S$, by the so called *average Hamming distance*. The use of Hamming distance based separation, rather than measures based on the l_1, l_2 or l_∞ norms, as is often the practice when the employing the vector space model, is suggested by the Boolean nature of our document model.

The *Hamming distance* between the vectors $\mathbf{x}[S] \in T[S]$ and $\mathbf{y}[S] \in F[S]$ is defined as $d_S(\mathbf{x}, \mathbf{y}) = \sum_{j \in S: x_j \neq y_j} 1$. The average Hamming distance between the sets $T[S]$ and $F[S]$ then is defined as

$$\Delta_{avg}(S) = \frac{1}{|T||F|} \sum_{\mathbf{x} \in T} \sum_{\mathbf{y} \in F} d_S(\mathbf{x}, \mathbf{y}). \qquad (1)$$

4 Ranking Functions

When selecting terms, ideally one should choose a group of terms, and not terms one by one, since co-occurrence of groups of terms carries typically more information then individual terms do. For this however, one would need to evaluate measure of "distinguishing power" (or simply *separation*) for sets of terms, greatly increasing the computational complexity. To keep up with our aims, we considered only methods choosing terms one by one. Furthermore, to be able to evaluate the true strength of the selected measures of separation, we choose terms independently (as opposed to sequential selection approaches).

According to the above objectives, for each term, i.e., for each index $i \in V$, the considered measures of separation are based only on the following four values

- $a_i \equiv$ the number of relevant documents containing the i^{th} term
- $b_i \equiv$ the number of irrelevant documents containing the i^{th} term
- $c_i \equiv$ the number of relevant documents which do not contain the i^{th} term
- $d_i \equiv$ the number of irrelevant documents which do not contain the i^{th} term

For each $i \in V$, the relationship between a_i b_i, c_i and d_i and the document collection is given by the following 2×2 contingency table

	$\mathbf{x} \in T$	$\mathbf{x} \in F$	
$x_i = 1$	a_i	b_i	$a_i + b_i = \theta_i$
$x_i = 0$	c_i	d_i	$c_i + d_i = \overline{\theta}_i$
	$a_i + c_i = \|T\|$	$b_i + d_i = \|F\|$	m

where the marginals θ and $\overline{\theta}_i$ represent the number of documents containing the i^{th} term and the number of documents which do not contain the i^{th} term respectively.

Obviously, the marginals $|T|$ and $|F|$ are constant for all terms while the marginals θ_i and $\overline{\theta}_i$ vary for each term. The total number of documents in the collection is $m = a_i + b_i + c_i + d_i$, which is obviously also a constant.

For the simplicity of notations, we shall view all ranking functions as functions of the four parameters $a = a_i$, $b = b_i$, $c = c_i$ and $d = d_i$, though clearly there are only two independent values among these. Also, for simplicity, we drop the index i. Each of the measures $\mu \in \{\alpha, \beta, \gamma, \delta, \rho\}$ considered below we view as a function $\mu = \mu(a, b, c, d)$, i.e. to the ith term ($i \in V$) we shall associate the value $\mu_i = \mu(a_i, b_i, c_i, d_i)$.

In [6] we analyzed how well different measures, based on the same four parameters associated to the terms, help in selecting terms yielding high separation between relevant and irrelevant documents. For each of the considered measures μ we ranked the terms by their corresponding μ_i value, selected the top k terms in this order, and then computed the average Hamming distance between relevant and irrelevant documents of the training set when those were represented only in terms of the selected k terms (repeated for various values of k). Furthermore, we also computed the average Hamming distance between the relevant and irrelevant documents of an independent test collection, using the same k terms for representing those documents. In this way we gained some insight into how stable and robust is the separating power of the selected terms. We did find several interesting results, surprising differences or similarities between different measures, etc. We refer the reader to [6] and the accompanying web-site for further details about these results.

In summary we found that quite a few measures provide very good, highly and robustly separating sets of terms. For instance, choosing any of the best performing measures, when we choose only the top 50 terms (out of the typically $20,000+$ terms of the collection), the average Hamming distance between the relevant and irrelevant documents in the training set is around 20 (i.e., for an average pair of relevant and irrelevant documents, there about 20 terms out of the chosen 50 which can be used to distinguish these two documents). What is even more promising is the fact that the same 50 terms separate the relevant and irrelevant documents of the independent test collection also about $19 - 20$-fold, i.e., the separation power of the same set degrades by less than

5% when we switch to the test set.

We have also noticed that the best measures were also very helpful in eliminating "noise". In other words, only a few "strange" terms were selected at the top of the ranked lists (which we could confirm by looking at the list of top ranked terms and the topic description). There were however differences in this respect between the measures, and we felt that it is better to rely on terms which are placed on the top of the list by several measures, not only by one of those.

Based on the above, we selected the following 5 such measures for our TREC experiment:

$$\alpha \quad = \quad \left| \frac{a}{a+c} - \frac{b}{b+d} \right| = \frac{|ad-bc|}{|T||F|} \tag{2}$$

is the absolute value of the difference between, the number of relevant-irrelevant document pairs in the training collection which provide evidence that the considered term is a good classifier of relevant documents, and the number of relevant-irrelevant document pairs which provide evidence that the considered term is a good classifier of irrelevant documents, normalized by the total number of relevant-irrelevant document pairs;

$$\beta = \frac{ad+bc}{(a+c)(b+d)} = \frac{ad+bc}{|T||F|} \tag{3}$$

is the total number of relevant–irrelevant document pairs correctly distinguished by the considered term, normalized by the total number of relevant-irrelevant document pairs in the training collection;

$$\gamma = \frac{ad}{(a+c)(b+d)} = \frac{ad}{|T||F|}$$

is an obvious variant of both α and *beta*;

$$\delta = \frac{|ad-bc|}{\sqrt{\theta\overline{\theta}|T||F|}} \tag{4}$$

is the absolute value of the *Pearson Product Moment Correlation* coefficient or simply the *correlation coefficient* of the occurrence of the considered term in the documents and the true classification of those documents.

Finally,

$$\rho = \frac{m(ad-bc)^2}{\theta\overline{\theta}|T||F|} \tag{5}$$

is the χ^2 statistic for the occurrence of the considered term in the documents and the true classification of those documents.

5 Training the Classifier

This section describes the feature selection method and the simple classifier used in the batch filtering sub-task of the filtering track.

The set of unique terms in the training set $T \cup F$ was ranked by each of the five ranking functions $\alpha, \beta, \gamma, \delta, \rho$ described in §4. Five intermediate feature sets, S_α, S_β, S_γ, S_δ, S_ρ, were constructed using the top ranking $K = 50$ terms of the corresponding ranking functions. Letting

$$\tilde{S} = S_\alpha \cup S_\beta \cup S_\gamma \cup S_\delta \cup S_\rho$$

we assigned a score $\psi \in \{1, \cdots, 5\}$ to each of the terms in $\tilde{S}$, defined as the number of sets S_ξ, $\xi \in \{\alpha, \beta, \gamma, \delta, \rho\}$ in which the term appeared. The final feature set S was constructed by selecting the $K = 50$ terms with the highest ψ scores.

Next, to each term in $i \in S$ we assigned the weight

$$\omega(i) = \frac{\frac{a_i+0.5}{a_i+c_i+1}}{\frac{b_i+0.5}{b_i+d_i+1}}$$

which can be seen to be the Yule-adjusted Bayesian weight of evidence, and to each document $\mathbf{y} \in T[S] \cup F[S]$ we assigned the score

$$\Omega(\mathbf{y}) = \sum_{j \in S} \log(\omega(i)) y_j.$$

That is, each document projected onto the selected feature set S is assigned a score equal to the sum of the logarithms of the Bayesian weights of evidence for the terms it contains.

The batch filtering task requires the definition of a static classification rule which specifies whether each

document in the test set should be considered relevant and retrieved, or irrelevant and ignored. The rule we utilized specifies that $\mathbf{y} \in T[S] \cup F[S]$ will be retrieved if and only if $\Omega(\mathbf{y}) \geq \tau$ for some $\tau \in \mathbb{R}$. In the classical pattern recognition literature such a threshold is chosen as the logarithm of the ratio of the a priori probabilities of the two classes (in the training set). This approach however does not take into account our objective, which is not simply finding as many as possible correct classifications, but to maximize the given utility function $TU11 = 2R - I$, where R is the number of correctly classified relevant documents in the training set, and I is the number of irrelevant documents, which were incorrectly classified as relevant. In our algorithm the threshold τ was selected so as to optimize the utility measure $TU11 = 2R - I$ over the training set.

6 Results

Let us remark first that in the collection of about 800,000 test documents only a small number (from few to a few hundreds) were relevant for any given topics. This implies that a trivial algorithm, not retrieving anything, is quite competitive since it does not collect mistakes for which the utility function assigned penalties. In fact many of the competing systems found it hard to beat this trivial algorithm.

In the TREC 2002 batch filtering track there were 100 topics specified, in two 50-50 groups. The first 50 topics, the so called *assessor judged topics*, consisted of information need formulated by human experts. The second 50 topics were formed by intersection from some very simple, automatically created categories.

On the intersection topics most systems performed miserably, typically not retrieving relevant documents in significant numbers – a not (yet) completely understood fact. Most systems did much better on the 50 assessor judged topics.

Our very simple approach performed quite well on both groups of topics, surely beyond our expectations, and placed itself in the middle of the competing 17 systems (and it performed better than the abovementioned trivial algorithm). This was partially due to the strict threshold selection, which helped us to eliminate part of the penalizing misclassifications. In the case of the intersection topics this implied that for 31 topics out of 50 we did not retrieve anything. On the 50 assessor based topics our system retrieved 698 documents in total, out of which 535 were relevant.

In summary, on the 50 *assessor judged topics* our TU11 score was less than the median of all competing systems fifteen times, equal to the median twenty times, greater than the median fifteen times, and never attained the maximum. On the 50 *intersection topics* our TU11 score was less than the median once, equal to the median six times, greater than the median forty-three times, and attained the the maximum twenty-one times.

7 Conclusion

The results of our batch filtering seem to provide some evidence that employing even very simple feature selection methods can indeed improve simple text classification methods and can be the foundation of a reasonably performing, fast system.

Future research will involve the running of experiments to determine the sensitivity of the feature set size K, the scoring threshold τ, as well as investigating additional ranking functions and their noise sensitivity.

References

[1] H. Almuallim and T. Dietterich. Efficient algorithms for identifying relevant features. In *Proceedings of the Ninth Canadian Conference on Artificial Intelligence*, pages 38–45, Vancouver, BC, 1992. Morgan Kaufmann.

[2] Andrei Anghelescu, Endre Boros, David Lewis, Vladimir Menkov, David Neu, and Paul Kantor. Rutgers filtering work at trec 2002: Adaptive and batch. In E. M. Voorhees and Lori P. Buckland, editors, *NIST Special Publication: SP 500-251 The Eleventh Text Retrieval Conference (TREC 2002)*. Department of Commerce, Na-

tional Institute of Standards and Technology, 2002.

[3] D. A. Bell and H. Wang. A formalism for relevance and its application in feature sebset selection. *Machine Learning*, 41:175–195, 2000.

[4] A. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 67:245–285, 1997.

[5] Endre Boros, Takashi Horiyama, Toshihide Ibaraki, Kazuhisa Makino, and Mutsunori Yagiura. Finding essential attributes from binary data. *Annals of Mathematics and Artificial Intelligence*, 2003. to appear.

[6] Endre Boros and David J. Neu. Rank based feature selection in information retrieval. Technical report, RUTCOR, Rutgers University, 2003. http://rutcor.rutgers.edu/˜neu/rbfs/index.html.

[7] C.K. Chow. Boolean functions realizable with single threshold devices. In *Proc. IRE,* **49**, pages 370–371, 1961.

[8] F.J. Banzaf III. Weighted voting doesn't work: A mathematical analysis. *Rutgers Law Review*, 19:317–343, 1965.

[9] G. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. In *Machine Learning: Proceedings of the Eleventh International Conference*, pages 121–129. Morgan Kaufmann Publisher, 1994.

[10] K. Kira and L. Rendell. The feature selection problem: Traditional methods and a new algorithm. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 129–134, Menlo Park, 1992. AAAI Press/The MIT Press.

[11] D. Koller and M. Sahami. Toward optimal feature selection. In *ICML-96: Proceedings of the Thirtenth International Conference on Machine Learning*, pages 284–292. Morgan Kaufmann, 1997.

[12] L.S. Shapley and M. Shubik. A method for evaluating the distribution of power in a committee system. *Amer. Polit. Sci. Rev.*, 48:787–792, 1954.

[13] E. M. Voorhees and Lori P. Buckland, editors. *NIST Special Publication: SP 500-251 The Eleventh Text Retrieval Conference (TREC 2002)*. Department of Commerce, National Institute of Standards and Technology, 2002.

[14] R.O. Winder. Chow parameters in threshold logic. *J. ACM*, 18:265–289, 1971.