

Exploring Feature Selection for Multi-Label Text Classification using Ranked Retrieval Measures

J. Scott Olsson
Dept. of Mathematics
University of Maryland
College Park, Maryland
olsson@math.umd.edu

Douglas W. Oard
College of Information Studies/UMIACS
University of Maryland
College Park, Maryland
oard@glue.umd.edu

ABSTRACT

While most classifier studies have focused on set-based evaluation measures, multi-label classification techniques that rank alternatives and then apply a threshold to make binary decisions can also be evaluated before thresholding. This can be done using well understood measures from ranked retrieval (R -precision in this case). Rank-based evaluation is first motivated by using a simple simulation to show that thresholding can introduce effects which obscure differences in the rank ordering of topics. The use of ranked retrieval measures for formative evaluation of multi-label classification is then demonstrated by exploring some techniques for combining evidence of term utility to improve feature selection for k -Nearest-Neighbor text classification. Because this ranked list evaluation framework greatly reduces the computational cost per method variant explored, we are able to investigate a large number of feature selection possibilities. Easily constructed combinations were found that proved to be more robust across a range of feature set sizes and that yielded statistically significant improvements in peak R -precision and microaveraged F_1 (a commonly reported set-based measure).

Categories and Subject Descriptors: H.3 [Information Storage and Retrieval]: Miscellaneous

General Terms: Experimentation, Measurement

Keywords: text classification, evaluation measures, feature selection

1. INTRODUCTION

Automatic text classification is a supervised machine learning problem which attempts to assign pre-defined categories to documents, based on a training set of labeled texts [9]. The problem is called multi-label if, naturally, multiple labels may be assigned to individual documents. This multi-label variant is especially important for information seekers, because the documents they hope to retrieve may generally be described in many different ways—that is, with many

different labels.

Multi-label text classification evaluation has traditionally focused on set-based measures, such as micro or macroaveraged F_1 [11]. These measures require classifiers to explicitly predict the set of topics for each test document. This prediction is normally accomplished by thresholding the classifier's score for each topic-document pair. Many classifier modifications could be evaluated before this thresholding, however, by considering the output ranked list of scores on topics. The evaluation of these lists may be conducted with well understood and widely used measure from the ranked retrieval community, such as R -precision. We show that using traditional, set based evaluations may inadvertently fail to detect significant improvements in experimental systems because they require the additional difficult task of score thresholding. Because many classification tasks eventually require strict topic assignment, we may consider a ranked list evaluation of the scores as an effort at problem decomposition. This decomposition is warranted by our focus on pre-thresholding system modifications, and is necessitated by the huge space over which we explore these variants.

To demonstrate this approach, we consider the problem of *feature selection*. In this application, our goal is to choose a small subset of the available features (i.e., words) on which to classify the documents. Feature selection can improve accuracy, by removing noisy or uninformative terms, and may greatly reduce the associated costs of classifying with many features. To capitalize on the unique strengths of different feature selection methods, we investigate several techniques by which one or more methods may be combined. That is, we explore feature selection combination methods.

Much work has been done comparing feature selection methods for text classification [14],[8] and the techniques (e.g., those based on the χ^2 statistic or information gain) have been widely adopted by the classification community. We might roughly divide the motivation for this adoption into an emphasis on evaluation gains and an emphasis on cost reduction. Some studies consider it a success to achieve sub-optimal evaluation results if the feature space is sufficiently reduced (these studies may even employ classifiers tending to monotonically improve an evaluation metric with the number of features, as with support vector machine (SVM) classifiers [10]). On the other hand, work emphasizing classification accuracy may require the careful removal of features to maximize an evaluation metric [7], particularly if the classifier is known to be susceptible to noise, as with the k -Nearest Neighbors (k NN) classifier.

For information retrieval, k NN is of special interest both because of its relatively low space/time complexity [15] and because it naturally handles the multi-label classification problem. While binary classifications must somehow be joined to produce a ranked list of labels for a user, k NN’s output is a ranked list of labels without adaptation. Binary classifiers such as SVMs must make strong label independence assumptions when applied to multi-label problems, and, because many binary subproblems must be combined for the multi-label problem, they scale poorly with the number of categories. Recent work has attempted to relax at least these independence assumptions [16],[17] although these efforts have incurred large additional costs in time. Even on problems with relatively small labelsets, k NN has been demonstrated to be competitive with SVMs [8] after careful feature selection. For very large labelsets (e.g., those that real life information seekers might hope to utilize in the wild), k NN’s low cost and implicit handling of label dependencies becomes increasingly attractive.

Our task then is to maximize k NN’s utility using fewer features. These features are obtained through combining several individual feature selection methods already in widespread use, which we refer to as *inputs*. To find strong combinations of our inputs, we must undertake an enormous investigation. This is greatly simplified by our ranked list exploration framework. We further motivate this approach in Section 2. We overview our input feature selection methods in Section 3. In Section 4, we introduce several possible methods for combining these inputs. Because the number of ways to combine input methods is extremely large, we conducted a preliminary search of the space of combination methods which we outline in Section 6. From this investigation we take the most promising combination methods and conduct a validation study on new data, which we present in Section 6.3. Finally, we conclude in Section 7.

2. EXPLORATION WITH RANKED LISTS

Set based measures such as macro- and micro-averaged F_1 are appropriate for evaluating *complete* classification tasks which require binary assignments of categories to documents. However, these complete classification systems are very often composed of two separate and far from trivial subproblems: producing category scores and thresholding these scores to produce category assignments. Thresholding is a difficult problem which has itself necessitated a great deal of research [12]. Consider, for example, the case of k NN, which naturally produces ranked lists of categories with scores: there is simply no reason *a priori* to expect score values for labels to be comparable across documents; accordingly, there is little theoretical (as opposed to empirical) justification for the choice of thresholds.¹ If, as is often the case, experimentation focuses on variants of the scoring subproblem, while evaluation is conducted on the completed task, we are forced to consider whether real system differences might be obscured by the thresholding component. To avoid these problems, we may simply evaluate the ranked lists rather than the sets produced from thresholding. As we’ll see,

¹Theoretically motivated thresholding strategies have been proposed and investigated (e.g., Pcut, which makes the distribution of categories similar in training and testing), but they tend to do worse on k NN than methods which simply tune thresholds with held out data [12].

if a system can produce better ranked lists, the improvements will translate into better sets post-thresholding. Our concern is whether we can *detect* these improvements when computational cost limits the number of feasible trials in a large investigation.

For our post-thresholding evaluation measure, we consider microaveraged F_1 . After thresholding, microaveraged F_1 is computed as the harmonic mean of precision and recall, $F_1 = \frac{2pr}{(p+r)}$. Precision, p , is defined as the proportion of all topic assignments which are correct. Recall, r , is the proportion of relevant topics actually assigned.

For ranked evaluation, we consider R -precision and mean average precision (MAP). R -precision is defined as simply the precision in the top R hypothesized categories for a document, where R is the number of categories which should be assigned to the document. This value is then averaged over all the testing documents. If a document has very many relevant topics, R -precision ensures we don’t judge its ranked list on the trivial problem of placing some of its many true labels in the top few positions. If a document has very few topics, R -precision ensures we don’t evaluate its ranked list on the futile attempt to fill many top positions with relevant labels when only a very few relevant labels actually exist. In other words, R -precision accounts for the variation in number of relevant topics across the test documents. To compute MAP, the precisions at each relevant topic in a document’s ranked list are averaged. The mean of these averages is then computed over the set of all documents.

To illustrate the possibility of thresholding obscuring classifier improvements, we conducted the following Monte Carlo experiments. Ranked lists were stochastically generated for 500 sets of 1000 documents having 100 categories, with each document having between one and five relevant labels. MAP, R -precision, and micro-averaged F_1 were computed for each set. For F_1 , we threshold the ranked lists to assign the top n categories, and use n such that F_1 is as large as possible.² This is not the best known thresholding strategy, but it suffices to illustrate our general concern. Figure 1 shows the pairwise plots of scores for each simulated set and the squared correlation r^2 between each evaluation measure. We observe the ranked list measures (MAP and R -precision) have significantly higher squared correlation than each ranked list method and F_1 . The discrepancy in squared correlation tells us that, as we would expect, F_1 is an inferior measure of the quality of the ranked list. The strong correlation between MAP and R -precision has been confirmed elsewhere [1],[2].

We then slightly extended this experiment by generating for each document the ranked list from a second classifier. We evaluate each classifier using each metric. If classifier one is deemed superior with respect to a measure, while classifier two is superior with respect to another, we say that the measure pair has produced a *reversal*. Table 1 shows the proportion of the 500 simulated sets which exhibited reversals for each measure pair. We see that the pair of ranked list measures tends to reverse significantly less frequently than pairs containing F_1 . This suggests evaluations using

²This is known as Rcut thresholding [12], the “R” of which should not be confused with the R in R -precision. The value of n is usually tuned using held-out data, but because our focus is on relative rather than absolute values we report only the peak value for F_1 under each experimental condition; results in real applications would be lower.

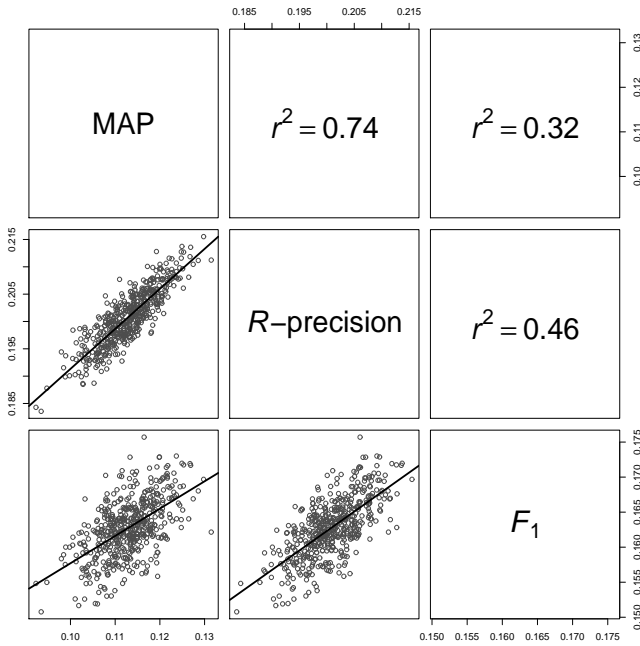


Figure 1: MAP, R -precision, and F_1 from 500 simulated sets of 1000 ranked lists of 100 categories. Squared correlation of measure scores, r^2 , is given above the diagonal.

Pair	MAP, R -precision	MAP, F_1	R -precision, F_1
Reversals %	0.194	0.314	0.352

Table 1: Proportion of 500 trials wherein two measures disagreed about which of two classifiers was superior.

F_1 are more likely to obscure improvements in the ranked lists when only a few trials for a method may be conducted.

Directly evaluating the ranked lists will be especially advantageous in cases where (1) the gains expected from modifying the classifier’s scoring component are relatively small, such that we may fear their detection being obscured by thresholding—particularly if so many method variants must be investigated that only a few exploration trials per variant are feasible (2) the methods investigated are themselves simple enough to warrant their adoption even if they offer only small (albeit reliable) improvements, (3) many method variants must be investigated, so that the here superfluous task of thresholding would drastically and fruitlessly increase the computational cost of method exploration.

We consider the problem of combining feature selection methods for text classification as an instructive example of such a scenario. Input feature selection methods are easily implemented and already widely available. Methods of combining these inputs are likewise very easy to implement, although quite numerous. To determine best combination methods, we must explore a large space of possible combinations. Feature selection improvements can naturally be evaluated before thresholding the ranked lists of scored categories, so there is no reason to encumber the method exploration by thresholding for each trial probe. As we shall see, we may combine these simple inputs in simple ways to reliably improve classifier scoring.

3. INPUT SELECTION METHODS

We focus on several widely used feature selection methods: document frequency thresholding, information gain, and χ^2 based methods such as χ^2_{\max} and χ^2_{avg} . Each method benefits from itself being easy to implement if not already available for use.

3.1 Document Frequency Thresholding

Document frequency (DF) is the number of documents in which a term occurs. In DF thresholding, terms with a DF below some threshold are discarded. Previous work has motivated this approach by suggesting that low DF terms are either not informative for category prediction or not influential for global performance [14]. This may be well motivated if rare terms are likely to be noise, although it violates the commonly held belief in information retrieval that rare terms are in fact more informative for information seekers than those which are common. DF thresholding has been found to be surprisingly effective, owing, it is suggested, to its generally strong correlation with information gain and χ^2 [14].

3.2 Information Gain

Information gain (IG) measures the number of bits of information obtained for category prediction by knowing whether a particular term is present or absent in a document [14]. The information gain $G(t)$ for a term t is defined as

$$G(t) = -\sum_{i=1}^m P(c_i) \log P(c_i) + P(t) \sum_{i=1}^m P(c_i|t) \log P(c_i|t) + P(\bar{t}) \sum_{i=1}^m P(c_i|\bar{t}) \log P(c_i|\bar{t}),$$

where $P(c_i)$ is the probability of category c_i and \bar{t} denotes the absence of term t for a classification problem with m categories.

3.3 χ^2 Statistics

The χ^2 statistic measures the degree of independence between a term t and a category c . It is defined as

$$\chi^2(t, c) = \frac{N(AD - CB)^2}{(A + C)(B + D)(A + B)(C + D)}$$

where A is the number of times t and c co-occur, B is the number of times t occurs without c , C is the number of times c occurs without t , D is the number of times neither c nor t occurs, and N is the number of documents in the training set. Since $\chi^2(t, c)$ is defined over the Cartesian product of terms and categories, for the purposes of feature selection, we must further conflate these category specific measures into a new statistic depending only on the term. This is commonly done by using either a weighted average

$$\chi^2_{\text{avg}} = \sum_{i=1}^m P(c_i) \chi^2(t, c_i)$$

or by taking the maximum value attained over all categories

$$\chi^2_{\max} = \max_{i=1}^m \{\chi^2(t, c_i)\}.$$

The χ^2_{\max} statistic is widely used [7] and considered to be a simple and effective feature selection strategy. It has been noted that the χ^2 statistic is unreliable for low frequency terms [14],[4], and performance gains have been observed by discarding low DF terms before computing $\chi^2_{\max}(t)$ [8].

4. COMBINATION APPROACHES

We now consider several methods for combining evidence from one or more individual input feature selection techniques. The combination approaches we consider are primarily from the text classification combination literature [5]. Many other combination strategies could be investigated (e.g., round robin selection of top ranked labels from input lists or weighted linear combinations of ranks or scores). The following techniques represent a sufficient sample to demonstrate gains may be achieved using feature selection combination.

4.1 Highest Rank

In highest rank (**HR**) combination, we give as each feature’s combined score the highest rank achieved in any of the input score lists. We expect this approach to be suitable if different input methods place different sets of informative features near the tops of their lists. At the same time however, this method will discount the negative information provided by an input; that is, if one input has high confidence that a feature is uninformative, it will be overruled by any other input which ranks the feature higher. Highest rank combination has previously been used to combine hypothesized labels (rather than features) in classification problems [5].

4.2 Lowest Rank

In lowest rank (**LR**) combination, we give as each feature’s combined score the lowest rank achieved in any of the input score lists.

4.3 Average Rank

In average rank (**AR**) combination, we give as each feature’s combined score the average rank achieved in all of the input score lists. We consider only the unweighted average of ranks, although it’s reasonable to expect this simplest approach to be sub-optimal.

4.4 Normalize then OR

Given two or more input vectors of feature scores, normalize them and, for each term, take the largest normalized feature score as the feature’s combined score. We may normalize the input feature scores in several different ways. If we normalize input score vectors by dividing each element by the vector’s largest element, then the **OR** effectively asks: for which input feature selection method did this term achieve a higher fraction of its largest observed score? We will refer to this approach as **DMOR** (divide by maximum then **OR**). If on the other hand, we normalize input score vectors by dividing each element by the vector’s (L_2) norm, then the **OR** asks: for which input feature selection method did this term achieve a higher fraction of the total achieved score across all the terms? We will refer to this approach as **DLOR** (divide by length then **OR**). Previous feature selection combination studies [8] used a normalize then **OR** combination approach, although to our knowledge they were limited to pairwise combinations using **DMOR**.

Note that this is computed on the feature scores rather than ranks; if instead ranks were used, the resulting sorted feature list would be identical to the highest rank combination. We’ll see that lowest rank combinations of input pairs tend to outperform **HR** combinations, suggesting that future work might investigate normalizing the input scores

and then choosing the *lowest* score from the normalized pair.

5. DF CUTTING

In many real world classification problems (e.g., naturally occurring text), we’d expect the lowest frequency terms to be predominantly noise. Moreover, many feature selection methods (particularly those of the χ^2 family) are known to be misled by infrequent terms [4]. For these reasons, and because the Zipfian distribution of term frequencies leads to significant time savings through the elimination of low **DF** terms, text classification studies often disregard them. We call this practice *DF cutting*, and say that, if all terms with **DF** less than or equal to C are ignored, that we are cutting at level C .

DF cutting differs subtly from **DF** thresholding (Section 3.1 above). The latter is a feature selection effort to select the few best terms, while the former is a noise reduction effort aimed at removing the few worst. **DF** cutting is done before computing feature selection scores, and so is aimed at mitigating the effects of infrequent terms on sensitive techniques such as χ^2_{\max} .

Although we would expect some feature selection methods to benefit more from aggressive **DF** cutting and others less, we restrict ourselves to considering only combinations of feature selection methods computed at the same cutting level (e.g., we don’t consider combinations of χ^2_{\max} at cut 5 and χ^2_{avg} at cut 0). We expect that such combinations would produce still better feature sets and may explore them in future work.

6. PRELIMINARY EXPERIMENTS

In these preliminary experiments, we leverage the task decomposition motivated in Section 2 to search the large space of possible feature selection combination methods. While we here conduct only a few trials per setting, this initial study included approximately 197 million classifications of documents (considering each group of methods, combination approach, feature set size, cutting level, and several independent trials). We will see that focusing on ranked list evaluation, thus reducing the risk of thresholding-induced reversals obscuring our observations, allows us to select promising combination candidates with only a few trials. Because the number of document classifications required grows linearly in the number of trials per setting, this greatly reduces the cost of our exploration. In addition, the cost is drastically reduced on account of our not tuning the thresholds needed for set-based evaluation.

6.1 Dataset and Classifier

Our preliminary dataset consists of 23,149 documents from the RCV1-v2 newswire corpus, as tokenized and stemmed by [7].³ RCV1-v2 is a collection of Reuter’s English news articles. Each article has one or more human assigned topics from a set of 101 categories. We used this topic set for the categorization task. We randomly partitioned the set 5 times, to produce five test/train splits on the data such that each training set contained 20% of the documents randomly assigned (i.e., about 4,600 documents). Each set of experiments was then run on each of these splits, for every

³In particular, we used the the training vector set from on-line appendix 13 of [7].

experimental setting (i.e., combination type, input methods, cutting level, and number of features).

It has been demonstrated that the k -Nearest Neighbors (k NN) algorithm is a strong performer in evaluations on the RCV1-v2 newswire corpus [7]. It has also been noted that k NN’s performance depends strongly on the careful selection of classification features. Feature selection studies have in fact demonstrated k NN outperforming support vector machine (SVM) classifiers in certain evaluation metrics (e.g., macroaveraged F_1) after careful feature selection was applied [8]. In these studies, while the performance of SVM classifiers tends to monotonically increase with the feature set size, k NN’s performance generally exhibits a strong peak at less than 10% of the features and then decreases slightly as the feature set increases to 100%. Because of this demonstrated potential, and because other classification methods (e.g., Naive Bayes) are not generally competitive [13] regardless of careful feature selection, we have restricted the present study to experiments using k NN.

We use a local implementation of k NN with symmetric Okapi term weighting,

$$w(tf) = \frac{tf}{0.5 + 1.5\left(\frac{dl}{avdl}\right) + tf}$$

where $w(tf)$ is the computed term weight, tf is the term frequency, dl is the length of the document in which the term occurs, and $avdl$ is the average document length. During classification, term weights are multiplied by their inverse document frequency (idf),

$$idf(t) = \log\left(\frac{N - df(t) + 0.5}{df(t) + 0.5}\right)$$

where $df(t)$ is the document frequency of term t and N is the total number of documents in training. The score for label ℓ on a test document with vector w_T is then computed as the sum of inner products between w_T and any of the k nearest document vectors which have label ℓ . That is,

$$score(w_T, \ell) = \sum_{i \in K_\ell} \sum_t w_T(t) w_i(t) idf(t),$$

where K_ℓ is the set of k nearest neighbors which have label ℓ assigned. For all experiments, we fixed k at $k = 100$.

6.2 Searching the Space

For each possible way to combine our input feature selection methods with our combination strategies, we ran experiments on our five preliminary splits. This entire process was repeated for three cutting levels (0, 1 and 5). For a given split, cutting level, and ordered list of features, experiments were run with k NN at feature set sizes: 50, 250, 500, 750, 1000, 1500, 2000, 2500, 3000, 4000, 5000, 10000, and all possible features. R -precision was computed and tabulated for each run.

6.2.1 A first look

Our primary motivation for feature selection is to improve classification accuracy, as compared with classifying using all features. To make a preliminary assessment as to whether each features selection method studied here does in fact consistently outperform the all-features case, we ran the initial experiments using the five data splits for every setting (that is, for every combination, combination method, feature set size, and cutting level).

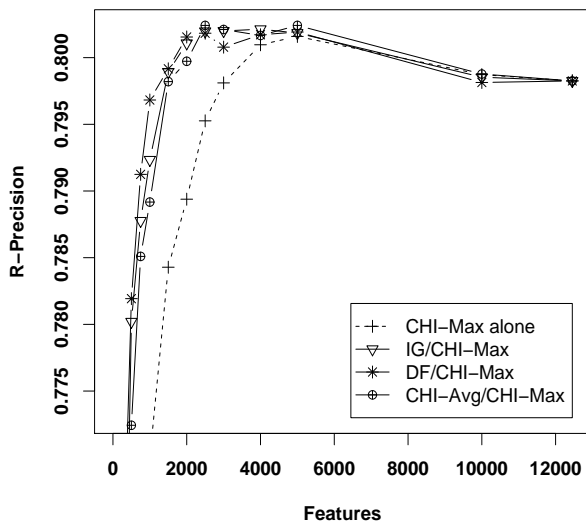


Figure 2: R -precision vs. feature set size for χ_{\max}^2 alone and all lowest rank (LR) combinations which include χ_{\max}^2 . Note that, for the combination methods, the curves are smoother (with flatter tops) and achieve peak R -precision for many fewer features.

Figure 2 shows a typical curve of R -precision vs. feature set size from the preliminary experiments. Accuracy initially improves as the feature set sizes increase. We’d like the R -precision to peak as early and as high as possible, and to assess which combinations do so most consistently. Moreover, we’d like to determine which methods outperform the all-features case over the broadest range of features. Such methods will minimize the risk of choosing a sub-optimal number of features.

Consider each of the five trials for a setting to be an independent trial with two possible outcomes: 1) the set of *all* features wins, 2) the selected set of features wins. By winning, we mean the feature set produces a higher R -precision. Now, if the winner were randomly decided (i.e., the coin was “fair”), then the probability of a particular feature selection strategy winning for each of the five trials would be $0.5^5 = .03125$ (i.e., it would be statistically significant at the level $\alpha = 0.05$). We therefore look for feature selection methods which, for a particular setting, beat the all-feature case in each of the five trials. Note that the all-features case for cutting levels 1 and 5 always produced better scores than at cutting level 0, so five wins will additionally demonstrate improvement over the most basic case—using all features with no cutting (a cutting level of zero). We visualize this analysis in Table 2 (tables for cutting level 0 and 5 were similar and so omitted to conserve space).

Combination methods appear to consistently beat the all-features case across a wider range of feature set sizes than non combination methods. In this sense, combination methods can be viewed as a risk reduction strategy for feature selection. Figure 2 shows a plot of R -precision vs. feature set size for χ_{\max}^2 features alone and each pairwise LR com-

Inputs	Method	1k	1.5k	2k	2.5k	3k	4k	5k	10k
DF	-								•
IG	-							•	•
χ_{\max}^2	-					•	•	•	•
χ_{avg}^2	-	•	•	•	•	•	•	•	•
IG/DF	HR								•
IG/ χ_{\max}^2	HR								•
IG/ χ_{avg}^2	HR								•
DF/ χ_{\max}^2	HR								•
DF/ χ_{avg}^2	HR								•
$\chi_{\max}^2/\chi_{\text{avg}}^2$	HR								•
IG/DF	LR								•
IG/ χ_{\max}^2	LR	•	•	•	•	•	•	•	•
IG/ χ_{avg}^2	LR	•	•	•	•	•	•	•	•
DF/ χ_{\max}^2	LR	•	•	•	•	•	•	•	•
DF/ χ_{avg}^2	LR	•	•	•	•	•	•	•	•
$\chi_{\max}^2/\chi_{\text{avg}}^2$	LR	•	•	•	•	•	•	•	•
IG/DF	AR								•
IG/ χ_{\max}^2	AR	•	•	•	•	•	•	•	•
IG/ χ_{avg}^2	AR	•	•	•	•	•	•	•	•
DF/ χ_{\max}^2	AR	•	•	•	•	•	•	•	•
DF/ χ_{avg}^2	AR	•	•	•	•	•	•	•	•
$\chi_{\max}^2/\chi_{\text{avg}}^2$	AR	•	•	•	•	•	•	•	•
IG/DF	DMOR								•
IG/ χ_{\max}^2	DMOR								•
IG/ χ_{avg}^2	DMOR								•
DF/ χ_{\max}^2	DMOR								•
DF/ χ_{avg}^2	DMOR								•
$\chi_{\max}^2/\chi_{\text{avg}}^2$	DMOR								•
IG/DF	DLOR								•
IG/ χ_{\max}^2	DLOR								•
IG/ χ_{avg}^2	DLOR								•
DF/ χ_{\max}^2	DLOR								•
DF/ χ_{avg}^2	DLOR								•
$\chi_{\max}^2/\chi_{\text{avg}}^2$	DLOR								•
DF/IG/ χ_{\max}^2	HR	•	•	•	•	•	•	•	•
DF/IG/ χ_{avg}^2	HR	•	•	•	•	•	•	•	•
DF/ $\chi_{\max}^2/\chi_{\text{avg}}^2$	HR	•	•	•	•	•	•	•	•
IG/ $\chi_{\max}^2/\chi_{\text{avg}}^2$	HR	•	•	•	•	•	•	•	•
DF/IG/ χ_{\max}^2	LR								•
DF/IG/ χ_{avg}^2	LR								•
DF/ $\chi_{\max}^2/\chi_{\text{avg}}^2$	LR								•
IG/ $\chi_{\max}^2/\chi_{\text{avg}}^2$	LR								•
DF/IG/ χ_{\max}^2	AR	•	•	•	•	•	•	•	•
DF/IG/ χ_{avg}^2	AR	•	•	•	•	•	•	•	•
DF/ $\chi_{\max}^2/\chi_{\text{avg}}^2$	AR	•	•	•	•	•	•	•	•
IG/ $\chi_{\max}^2/\chi_{\text{avg}}^2$	AR	•	•	•	•	•	•	•	•
DF/IG/ χ_{\max}^2	DMOR								•
DF/IG/ χ_{avg}^2	DMOR								•
DF/ $\chi_{\max}^2/\chi_{\text{avg}}^2$	DMOR								•
IG/ $\chi_{\max}^2/\chi_{\text{avg}}^2$	DMOR								•
DF/IG/ χ_{\max}^2	DLOR								•
DF/IG/ χ_{avg}^2	DLOR								•
DF/ $\chi_{\max}^2/\chi_{\text{avg}}^2$	DLOR								•
IG/ $\chi_{\max}^2/\chi_{\text{avg}}^2$	DLOR								•
DF/IG/ $\chi_{\max}^2/\chi_{\text{avg}}^2$	HR	•	•	•	•	•	•	•	•
DF/IG/ $\chi_{\max}^2/\chi_{\text{avg}}^2$	LR								•
DF/IG/ $\chi_{\max}^2/\chi_{\text{avg}}^2$	AR	•	•	•	•	•	•	•	•
DF/IG/ $\chi_{\max}^2/\chi_{\text{avg}}^2$	DMOR								•
DF/IG/ $\chi_{\max}^2/\chi_{\text{avg}}^2$	DLOR								•

Table 2: Dot sizes depict the number of times a feature selection approach, for some number of features, beat the all features case in our preliminary (cutlevel 1) experiments. $\cdot=1, \bullet=2, \circ=3, \ominus=4, \odot=5$.

Inputs	Method	50	250	.5k	.75k	1k	1.5k	2k	2.5k	3k	4k	5k
DF	-											
IG	-											
χ_{\max}^2	-	•	•	•	•	•	•	•	•	•	•	•
χ_{avg}^2	-	•	•	•	•	•	•	•	•	•	•	•
IG/DF	HR	•	•	•	•	•	•	•	•	•	•	•
IG/ χ_{\max}^2	HR	•	•	•	•	•	•	•	•	•	•	•
IG/ χ_{avg}^2	HR	•	•	•	•	•	•	•	•	•	•	•
DF/ χ_{\max}^2	HR	•	•	•	•	•	•	•	•	•	•	•
DF/ χ_{avg}^2	HR	•	•	•	•	•	•	•	•	•	•	•
$\chi_{\max}^2/\chi_{\text{avg}}^2$	HR	•	•	•	•	•	•	•	•	•	•	•
IG/DF	LR	•	•	•	•	•	•	•	•	•	•	•
IG/ χ_{\max}^2	LR	•	•	•	•	•	•	•	•	•	•	•
IG/ χ_{avg}^2	LR	•	•	•	•	•	•	•	•	•	•	•
DF/ χ_{\max}^2	LR	•	•	•	•	•	•	•	•	•	•	•
DF/ χ_{avg}^2	LR	•	•	•	•	•	•	•	•	•	•	•
$\chi_{\max}^2/\chi_{\text{avg}}^2$	LR	•	•	•	•	•	•	•	•	•	•	•
IG/DF	AR	•	•	•	•	•	•	•	•	•	•	•
IG/ χ_{\max}^2	AR	•	•	•	•	•	•	•	•	•	•	•
IG/ χ_{avg}^2	AR	•	•	•	•	•	•	•	•	•	•	•
DF/ χ_{\max}^2	AR	•	•	•	•	•	•	•	•	•	•	•
DF/ χ_{avg}^2	AR	•	•	•	•	•	•	•	•	•	•	•
$\chi_{\max}^2/\chi_{\text{avg}}^2$	AR	•	•	•	•	•	•	•	•	•	•	•
IG/DF	DMOR	•	•	•	•	•	•	•	•	•	•	•
IG/ χ_{\max}^2	DMOR	•	•	•	•	•	•	•	•	•	•	•
IG/ χ_{avg}^2	DMOR	•	•	•	•	•	•	•	•	•	•	•
DF/ χ_{\max}^2	DMOR	•	•	•	•	•	•	•	•	•	•	•
DF/ χ_{avg}^2	DMOR	•	•	•	•	•	•	•	•	•	•	•
$\chi_{\max}^2/\chi_{\text{avg}}^2$	DMOR	•	•	•	•	•	•	•	•	•	•	•
IG/DF	DLOR	•	•	•	•	•	•	•	•	•	•	•
IG/ χ_{\max}^2	DLOR	•	•	•	•	•	•	•	•	•	•	•
IG/ χ_{avg}^2	DLOR	•	•	•	•	•	•	•	•	•	•	•
DF/ χ_{\max}^2	DLOR	•	•	•	•	•	•	•	•	•	•	•
DF/ χ_{avg}^2	DLOR	•	•	•	•	•	•	•	•	•	•	•
$\chi_{\max}^2/\chi_{\text{avg}}^2$	DLOR	•	•	•	•	•	•	•	•	•	•	•

Table 3: Dot sizes graphically depict the number of times (out of five independent trials) the experiment at cutting level 5 achieved a higher R-precision than its cutting level 1 counterpart, where $\cdot=1, \bullet=2, \circ=3, \ominus=4, \odot=5$. Note especially the methods incorporating χ_{\max}^2 , which consistently improve with increased cutting level, as we would expect from their known sensitivity to low DF terms.

bination which includes χ_{\max}^2 . Note that χ_{\max}^2 alone, while achieving approximately the same peak R-precision, does so with many more features; all three LR combinations achieve peak R-precision at a lower feature set size, and their respective graph is considerably less pointed. The implication here is that, if one is going to use held out data to tune the number of features, using a combination approach (even if it does not outperform χ_{\max}^2 alone for some number of features) will often reduce the risk of choosing a sub-optimal feature set size (i.e., of falling off the performance peak).

Somewhat surprisingly, Table 2 shows us that pairwise LR combinations outperform their HR counterparts, while, for combinations of three or more inputs, the situation is reversed. This is similarly repeated for both the cutting level 0 and 5 experiments. Future work might explore this phenomena in more depth.

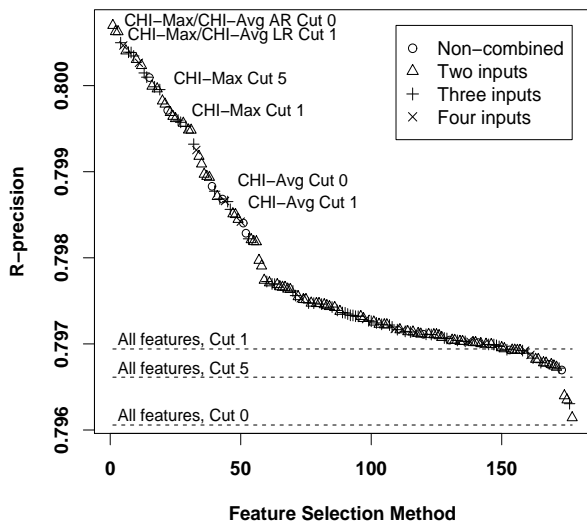


Figure 3: Peak R -precision averaged over the preliminary experiments for each method investigated. The top few non-combination and combination methods are labeled.

Table 3 explores our preliminary results in another way, asking the question: how often does a cutting level 5 experiment outperform its cutting level 1 counterpart? Note especially the consistent gains at increased cutting level in methods incorporating χ^2_{\max} . This conforms to our understanding that χ^2 is unreliable for low frequency terms [4].

Finally, consider Figure 3, which shows the R -precision averaged over the preliminary trials for each method investigated. The top few non-combination and combination methods are labeled. Not surprisingly, the best performing non-combination methods are based on χ^2 , as are the best performing combination methods. Note that the non-combined methods appear to benefit from more aggressive cutting than the best combination methods. We see performance improvements for the all-features case with low cutting, while performance suffers if the cutting is too high.

6.3 Validation Experiments

The preliminary experiments reported in Section 6.2.1 were motivated by an effort to begin searching the immense space of possible feature selection combination methods. However, because these initial probings included so few experiments for any particular setting, there is a danger that our conclusions may not generalize to new data. For this reason, we conducted a second set of validation experiments, now including only those combination methods suggested by the preliminary investigation, but with many more trials to ensure our results are statistically significant.

Our new data is a set of 200,000 RCV1-v2 documents, incorporating no documents from our preliminary study. We partitioned this new data into 20 disjoint sets of 10,000 documents each, before further dividing each of these in half to produce 20 pairs of testing and training sets of 5,000 documents. Although this data is completely separate from that

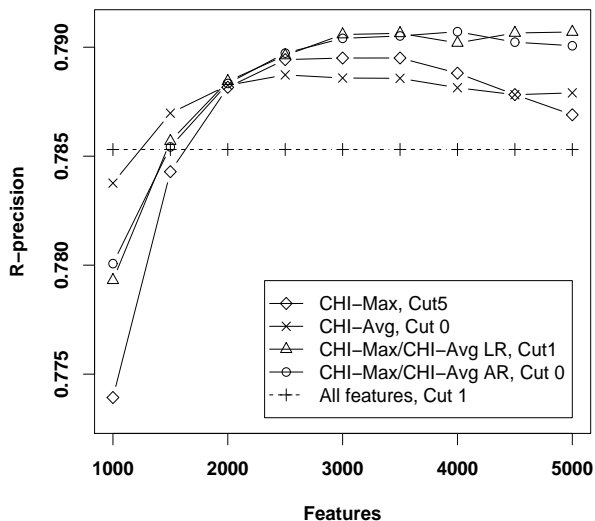


Figure 4: R -precision vs. feature set size averaged over the twenty validation experiments, as outlined in Section 6.3.

of the preliminary experiments, we chose to train on approximately the same *number* of documents as in the preliminary investigation; we hoped this would allow our insights to map over fairly from the preliminary investigation, since, for example, we would expect a cutting level of 5 to have similar effects on roughly equally sized training sets, and different effects on training sets of different sizes. We again consider the 101 topic categories for our classification task.

For each testing/training pair we conducted experiments using k NN with the top 1000, 1500, 2000, 2500, 3000, 3500, 4000, 4500, and 5000 features from each feature selection method. This range of feature sizes was sufficiently broad to capture the rise and peak of R -precision for each of the methods we considered.

We took the combination and non-combination methods from our preliminary investigation with the highest average peak R -precision. These were: (1) χ^2_{\max} at cutting level 5, (2) χ^2_{avg} with no cutting, (3) the all-features case at cutting level 1, (4) the LR combination of χ^2_{\max} and χ^2_{avg} at cutting level 1, and (5) the AR combination of χ^2_{\max} and χ^2_{avg} at cutting level 0. Methods (1)-(3) represent widely used methods which are known to be strong performers—particularly (1) and (2), while methods (4) and (5) represent the most strongly suggested combination approaches from our preliminary study.

Figure 4 plots the averaged R -precision at each feature set size across the 20 experiments. As before, we note that the combination methods peak higher and over a broader range of features. We observe that χ^2_{avg} overcomes the all-features case earlier than the combination methods. This is confirmed by Figure 5, which shows how many trials in 20 each feature selection method beat the all-features case. Recall however that we selected combination approaches from the preliminary study to attain a highest peak R -precision, not an earliest rise.

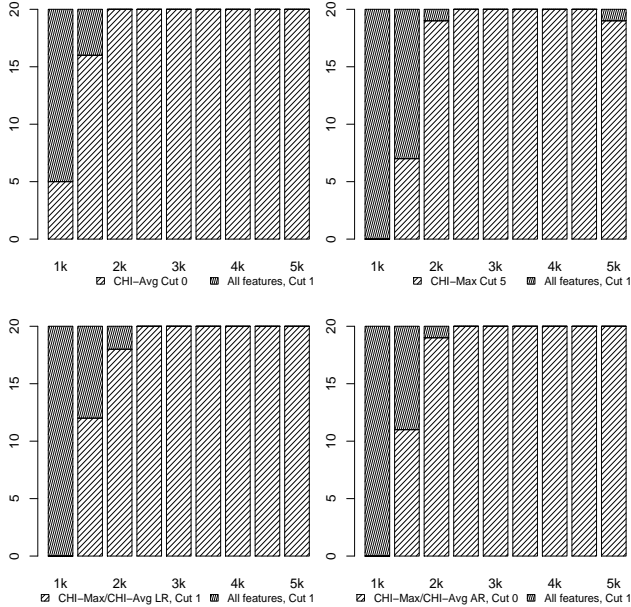


Figure 5: Bars depict how often each method achieved the higher R -precision in 20 trials, comparing the all-features case and each of the feature selection methods.

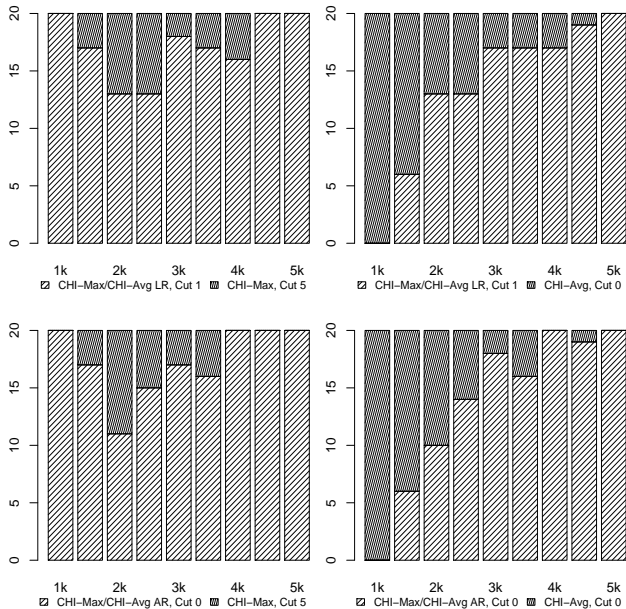


Figure 6: Bars depict how often each method achieved the higher R -precision in 20 trials, comparing combination and non-combination approaches.

To test for statistically significant improvement between a pair of feature selection methods, we may employ the non-parametric Fisher Sign test, comparing paired R -precisions for each of the 20 trials. Note that for 20 trials, 15 wins will be statistically significant, having a p-value of 0.021. To compare each feature selection method pair, Figure 6 de-

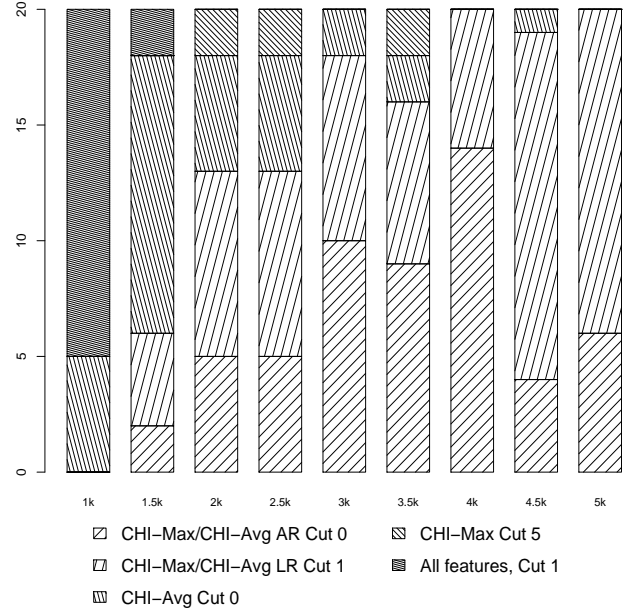


Figure 7: Bars depict how often each method achieved the highest R -precision in the 20 splits, at each feature set size.

picts how often each method in a pair achieved the highest R -precision, for each of the set sizes. We observe that the combination methods improve over non-combination methods at a statistically significant level over the majority of feature set sizes. If we count how often each method achieved the very highest R -precision for each feature set size, we see that, for most set sizes, the combination methods dominate. This is illustrated in Figure 7.

We have seen that the combination methods achieved higher R -precision over the majority of individual set sizes, but recall that we selected methods from our preliminary study to attain a highest *peak* R -precision. Figure 8 shows how often each method in each pair obtained the highest peak over all set sizes. Because many complete tasks require thresholding to produce label sets, we also report here microaveraged F_1 . We see that combination methods beat out the non-combination methods for peak R -precision, at statistically significant levels, for every pair observed. The improvement is similarly clear in the F_1 results, although we note that $(F_1, R$ -precision) reversals occurred on a small number of the validation partitions. While our exploration was facilitated by rank-based evaluation, we see here that the techniques discovered retain their improvements even after thresholding.

7. CONCLUSION

One hallmark of the engineering approach that underlies much of what we do is decomposition of complex problems into simpler ones with well characterized input-output behavior. These problems may be more easily optimized individually than they would be together. The approach described in this paper fits that perspective well. While more tightly coupled approaches to the multi-label assignment problem are possible, decomposing the problem into ranking followed by selection has proven to be reasonably effective.

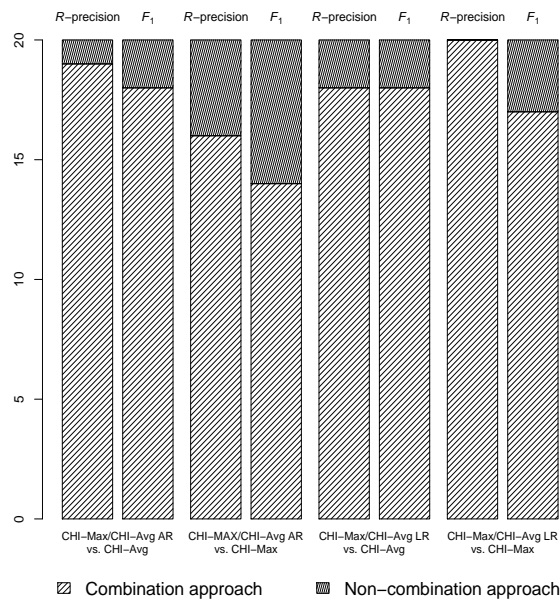


Figure 8: Bars depict how often each method achieved the highest R -precision and F_1 for a split, considering all feature set sizes.

tive and therefore quite popular. Moreover, the R -precision measure that we selected seems well matched to rank-based thresholding strategies, as the comparable results with F_1 in our last study illustrate. The principal advantage of this approach to formative evaluation is that by partitioning the problem it becomes possible to rapidly explore a much larger range of alternatives than would otherwise be possible. So long as empirical techniques figure prominently in text classification research, this will remain a useful capability.

8. ACKNOWLEDGMENTS

This work has been supported in part by (RFBR).

9. REFERENCES

- [1] J. A. Aslam, E. Yilmaz, and V. Pavlu. A geometric interpretation of R -precision and its correlation with average precision. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 573–574, 2005. ACM Press.
- [2] C. Buckley and E. M. Voorhees. Evaluating evaluation measure stability. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 33–40, New York, NY, USA, 2000. ACM Press.
- [3] C. Buckley and E. M. Voorhees. Retrieval evaluation with incomplete information. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 25–32, New York, NY, USA, 2004. ACM Press.
- [4] T. Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74, 1993.
- [5] T. K. Ho, J. J. Hull, and S. N. Srihari. Decision Combination in Multiple Classifier Systems. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(1):66–75, 1994.
- [6] J. Kekäläinen and K. Järvelin. Using graded relevance assessments in IR evaluation. *J. Am. Soc. Inf. Sci. Technol.*, 53(13):1120–1129, 2002.
- [7] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. RCV1: A New Benchmark Collection for Text Categorization Research. *J. Mach. Learn. Res.*, 5:361–397, 2004.
- [8] M. Rogati and Y. Yang. High-performing feature selection for text classification. In *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*, pages 659–661, New York, NY, USA, 2002. ACM Press.
- [9] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.
- [10] J. Yan, N. Liu, B. Zhang, S. Yan, Z. Chen, Q. Cheng, W. Fan, and W.-Y. Ma. OCFS: Optimal orthogonal centroid feature selection for text categorization. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 122–129, New York, NY, USA, 2005. ACM Press.
- [11] Y. Yang. An Evaluation of Statistical Approaches to Text Categorization. *Information Retrieval*, 1(1-2):69–90, 1999.
- [12] Y. Yang. A study of thresholding strategies for text categorization. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 137–145, New York, NY, USA, 2001. ACM Press.
- [13] Y. Yang and X. Liu. A re-examination of text categorization methods. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 42–49, New York, NY, USA, 1999. ACM Press.
- [14] Y. Yang and J. O. Pedersen. A Comparative Study on Feature Selection in Text Categorization. In *ICML '97: Proceedings of the Fourteenth International Conference on Machine Learning*, pages 412–420, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.
- [15] Y. Yang, J. Zhang, and B. Kisiel. A scalability analysis of classifiers in text categorization. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 96–103, New York, NY, USA, 2003. ACM Press.
- [16] K. Yu, S. Yu, and V. Tresp. Multi-label informed latent semantic indexing. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 258–265, New York, NY, USA, 2005. ACM Press.
- [17] S. Zhu, X. Ji, W. Xu, and Y. Gong. Multi-labelled classification using maximum entropy method. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 274–281, New York, NY, USA, 2005. ACM Press.