Methods for Learning Classifier Combinations: No Clear Winner

Dmitriy Fradkin
DIMACS
Piscataway, NJ
dfradkin@paul.rutgers.edu

Paul Kantor
DIMACS
Piscataway, NJ
kantor@scils.rutgers.edu

ABSTRACT

This work compares two approaches to finding effective topic-independent classifier combinations. We suggest a new federated approach and compare it against the global approach. Our results indicate that the relative effectiveness of these approaches depends on the measure used to evaluate them. We suggest explanations for these results.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; D.2.8 [**Software Engineering**]: Metrics—*performance measures*

General Terms

Experimentation, Measurement, Performance

Keywords

Text Classification, Classifier Combination

1. INTRODUCTION

Previously, there were two approaches to classifier combination. "Local" combination is topic-specific, and is computed and evaluated on a single topic. "Global" combination is computed on one set of topics and evaluated on another. Our new approach for constructing topic-independent classifier combinations falls somewhere between. We compute local combinations on individual topics, but combine them together to apply to *other* topics. We call our approach "federated".

We compare the federated approach with the individual classifiers and with the global approach. We analyze several local linear combination methods.

We find that the choice of the better approach, and of the best local rule, dependends on the measures used to compare them. This highlights importance of choosing a measure that is appropriate to the real task.

SAC '05, March 13-17, 2005, Santa Fe, New Mexico, USA

2. RELATED WORK

Existing work on combining heterogeneous classifiers for information retrieval is widely varied in measures, goals and tasks.

Bartell et. al. suggested a "global" linear combination of classifiers in 1994 [1], optimizing weights on all training queries simultaneously and applying the combination to the test queries in a batch mode. Optimization was based on minimizing Guttman's Point Alienation criterion with a conjugate gradient descent procedure, while the performance was evaluated on the basis of the average precision. Note that the criterion being optimized was not the measure used to evaluate the combination.

Hull et. al. (1996) [4] evaluated "local" classifier combinations for document filtering. They examined averaging strategies (based on probabilities and log-odds), linear regression, logistic regression and grid search on a predefined grid, and aimed to optimize a TREC-style utility measure.

Larkey and Croft (1996) [6] examined "local" combinations of kNN method with relevance feedback and Bayesian independence classifiers for categorizing patient discharge summaries. Here the categories are the ICD-9 codes. The methods are linear combinations of ranks or scores for all pairs involving kNN and one 3-way combination. The coefficients were found using grid search. All combinations performed better than individual classifiers, with the 3-way combination having the best results.

Yang et. al. (2000) [9] examined equal weight combinations of classifiers for query detection and tracking, noting that it produces better results than individual classifiers.

Generalization of classifier combination methods were suggested by Lam and Lai (2001) [5] and Bennett et. al. (2002) [2]. [5] uses category-specific features to model error rates of different classifiers on a validation set, and then picks the best classifier for each query to be used on the test set. This method performs better than individual classifiers. In [2], "reliability indicators" (novel document-specific features, not used in training, and some representation-specific and classifier-specific features) are used in addition to scores as input to a meta-classifier (decision tree or linear SVM). The resulting hierarchical systems perform better than individual classifiers and simple combination methods.

3. DATA

We used RCV1-v2 corpus [7] which is completely judged for the set of 103 queries. The training set (23,149 documents) was used to train individual classifiers. A test set of 4,060 documents was randomly selected from remaining

781,265 for evaluating classifier combinations. The relative effectiveness of the individual classifiers on this set is similar to that on the whole corpus. We used 99 queries having at least one relevant document in the test set.

4. MEASURES

We used a T11SU measure (TREC-11 [8]). We fix a query. Let T be the test set. Let T^+ be the set of all positive documents for the query in the test set. Let D be a set of all submitted (retrieved) documents, and let D^+ be the set of positive documents retrieved, and D^- be the set of nonrelevant documents retrieved. $|S|$ denotes the number of elements in a set S. Then, the T11SU measure is defined as: $\text{T11SU} = \frac{\max(\text{T11NU}, -0.5)+0.5}{1.5}$, where $\text{T11NU} = \frac{2|D^+|-|D^-|}{2|T^+|}$.

F1 measure is sometimes used to evaluate performance of a system. One of our single systems was trained to optimize that measure: $\text{T11F} = \frac{2|D^+|}{|D|+|T^+|}$.

5. COMBINATIONS OF CLASSIFIERS

For a given query, let L be the training set, L^+ - the set of positive training documents, and L^- - the set of nonrelevant training documents. Let $y(d,q) = \{0,1\}$ be relevance judgment (label) for a document d in regard to query q. The dot product is denoted by bold-font parenthesis: $(\mathbf{)}$.

A decision rule $C(d,q)$ produces a number 1 or 0 for a document-topic pair d and q. The value of 1 means that the document should be sent to the user interested in topic q, while the value of 0 means that the document should be ignored for q.

5.1 Document Representation

Each document is represented by a real-valued vector, with entries corresponding to terms. Denote the "raw term frequency" of a term t with respect to a document or a query by $f'(t,d)$. This is the number of times that term occurs in this document. The entry for each term t is usually derived from $f'(t,d)$. Some methods use $f'(t,d)$ itself. Other methods use a function of "raw term frequency" such as:

$$f(t,d) = \begin{cases} 1 + log(f'(t,d)), & f'(t,d) > 0 \\ 0, & f'(t,d) = 0 \end{cases} \quad (1)$$

Based on the training corpus it is possible to derive the document frequency, $w'(t)$ - the number of documents that contain t. The so-called "term frequency" (f or f') is usually weighted by some function of the term's document frequency. The intuition behind this approach is that the most frequent terms are not particularly useful in classification and should be weighted down, while rarer terms should have more weight. One particular weighting formula is:

$$w(t) = log(\frac{1+|L|}{1+w'(t)}) \quad (2)$$

5.2 Individual Classifiers

We make use of three classifiers in our experiments.

5.2.1 Bayesian Logistic Regression Classifier

One of the methods used is a Bayesian Logistic Regression classifier [3]. $f'(t,d)w(t)$ is the weight corresponding to term t is representation of document d.

Let l be the logit transformation:

$$l(x) = \frac{e^x}{1+e^x} \quad (3)$$

Then, the score of a document t can be computed as:

$$r_B(d,q) = l((d,\gamma_q)). \quad (4)$$

Here γ_q is a (dual) parameter vector chosen during the training stage to minimize:

$$\sum_{d \in L} \log((2y(d,q)-1)(d,\gamma_q)) + \log P(\gamma_q), \quad (5)$$

where P is a prior distribution of γ_q. (The reason $(2y(d,q)-1)$ is used is to give nonrelevant examples negative weight). In our case, independent normal priors for each term are used: $P = N(0, 0.01)$.

The decision rule can now be formulated as:

$$C_B(d,q) = sign(r_B(d,q) - \tau_q), \quad (6)$$

where τ_q is a threshold chosen during the training stage to maximize the F1 measure.

5.2.2 kNN Classifier

Another classifier used is a traditional kNN classifier with $k = 384$. The term representation is standard *tf-idf*: $f(t,d)w(t)$. The document vectors d are normalized to have length 1. Similarity, the inner product, is equivalent to the cosine of the angle between them.

A document is scored for a query q as follows:

$$r_N(d,q) = \sum_{i \in N(d)} y(i,q), \quad (7)$$

where $N(d)$ is the set of k nearest neighbors of d.

The traditional kNN decision function is:

$$C_N(d,q) = sign(r_N(d,q) - \frac{k}{2}), \quad (8)$$

5.2.3 Rocchio Classifier

The third method is a Rocchio classifier. The terms are represented by *tf-idf*: $f(t,d)w(t)$. This particular version of Rocchio is constructed as follows:

The document vectors d are normalized to have length 1. The query vector is represented as:

$$q = \frac{1}{|L^+|} \sum_{d \in L^+} d - \frac{1}{|L^-|} \sum_{d \in L^-} d. \quad (9)$$

A score $r_R(d,t)$ is computed as:

$$r_R(d,q) = (d,q). \quad (10)$$

The threshold τ_q is chosen to give best results on the training set.

The final decision rule is:

$$C_R(d,q) = sign(r_R(d,q) - \tau_q), \quad (11)$$

5.3 Score Normalization

As we have seen, each classifier s assigns every document-query pair (d,q) a score $r_s(d,q)$ that is indicative of the degree of similarity between the document and the query.

In order to combine scores, we fit all the scores on the test set to the interval [0,1]. This is achieved in the following

way:

$$x_s(d,q) = \frac{r_s(d,q) - \min_{d' \in T} r_s(d',q)}{\max_{d' \in T} r_s(d',q) - \min_{d' \in T} r_s(d',q)}. \quad (12)$$

5.4 Combination Framework

We examine only linear combination decision rules. They can be described as having the following form:

$$C_F(d,q) = sign(\sum_{j=1,\ldots,l} \beta_j x_j(d,q) - \tau_q), \quad (13)$$

where β is an l-dimensional vector of weights and τ is a threshold. Below we discuss methods for finding β and τ.

5.4.1 Computing query-specific parameters

Assume that for a query q we have represented each document as a vector of scores from individual systems for a given query t: $x_i = \{x_{ij}\}$ - jth coordinate of a vector x_i is a (normalized) score assigned to document i by system j (for query q). In the discussion in this section we will omit subscripts for the query, with the understanding that all the computations are query specific. Let y be an $|T| \times 1$ vector of relevance judgments: zeroes and ones.

The first of our combination methods attempts to minimize a squared error criterion:

$$E_1 = \sum_{i=1,\ldots,n} (y_i - \beta \cdot x_i)^2 \quad (14)$$

over all possible parameter vectors $\beta \in \Re^l$ The best solution can be found using linear algebra and notion of the pseudoinverse. Once the vector β is found, it is normalized to have unit length. Since this method involves solving a linear regression problem, we refer to it as $L1$.

Our second combination method is $L1$ with an intercept. The criterion minimized over β and δ is:

$$E_2 = \sum_{i=1,\ldots,n} ({y^t}_i - \beta \cdot x_i - \delta)^2. \quad (15)$$

Because of δ, lines that do not go through the origin may be explored. This method usually leads to a different vector β from the method $L1$. Here β is also normalized to have unit length. We refer to this method as $L2$. While the $L1$ method has been explored in the literature ([1], [4]), we have not seen references to $L2$ in any previous work.

The third method involves fitting a logistic regression model to the relevance vector y and the document score vectors x_i. This requires maximizing over β the following criterion:

$$E_3 = \sum_{i=1,\ldots,n} \{y_i log(p(y_i = 1|x_i;\beta)) + (1-y_i) log(1 - p(y_i = 1|x_i;\beta))\}, \quad (16)$$

where

$$p(y_i = 1|x_i;\beta) = l(\beta \cdot x_i + \delta). \quad (17)$$

Note that

$$p(y_i = 0|x_i;\beta) = 1 - l(\beta \cdot x_i + \delta), \quad (18)$$

and so

$$log(\frac{p(y_i = 1|x_i;\beta)}{p(y_i = 0|x_i;\beta)} = \beta \cdot x_i + \delta, \quad (19)$$

Since log is a monotone transformation, we can simply use $\beta \cdot x_i$ as an indicator of relevance of x_i.

The last method for local combination that we discuss is a "centroid" method. A centroid (or a mean) $\overline{x_i}$ of a finite set of points $\{x_i | i = 1, \ldots, n\}$ is computed as

$$\overline{x_i} = \frac{1}{n} \sum_{i=1,\ldots,n} x_i. \quad (20)$$

We first compute the centroids $\overline{x_{i+}}$ and $\overline{x_{i-}}$ of the relevant and the nonrelevant documents. We then define

$$\beta = \frac{\overline{x_{i+}} - \overline{x_{i-}}}{||\overline{x_{i+}} - \overline{x_{i-}}||}. \quad (21)$$

β is the normalized vector connecting the centroids of relevant and nonrelevant documents. If positive and negative scores have Normal distribution with equal covariance matrices, the optimal (Bayes) decision rule would have the form of $sign(\beta \cdot x_i - \tau)$ for some utility-dependent threshold τ. In reality the distribution of the scores need not be Normal, but this may be a good heuristic. This method has not been used for classifier combination for information retrieval.

In each of the above four methods, once the vector β is found we compute a set of scores $\beta \cdot x_i$ and find a threshold τ optimizing the T11U measure.

5.4.2 Combining classifiers across queries

This section describes the federated approach, where the parameters of query-specific combinations are used to create a rule that can perform well on other topics.

We have a set of training queries $q_1, ..., q_m$. For each query q_i we have the best (as computed by one of the methods discussed in Section 5.4.1) vector of coefficients β_i and threshold τ_i. Notice that τ_i is directly computed based on documents in q_i and on β_i to maximize some utility function. So we can write $\tau_i = \psi(q_i, \beta_i)$. Also notice that for any scalar $a > 0$, $\psi(q_i, a\beta_i) = a\psi(q_i, \beta_i) = \alpha\tau_i$, since scaling β_i results in the scaling of the scores for documents, but their order doesn't change. And thus scaling τ_i by the same factor leads to the same utility.

The problem now is to come up with a single vector of coefficients β^* and threshold τ^* to be used on test queries.

We suggest averaging query-specific weights in order to obtain β^*. In other words:

$$\beta^* = \frac{1}{m} \sum_{j=1,\ldots,m} (\beta_i), \quad (22)$$

where m are the queries in the training set.

Then τ^* can be computed as follows:

$$\tau^* = \frac{1}{m} \sum_{j=1,\ldots,m} (\tau_j) = \frac{1}{m} \sum_{j=1,\ldots,m} (\psi(q_j, \beta_j)) \quad (23)$$

We refer to this method as M1.

We have explored another possibility, M2. This method computes β^* in the same way as M1, but it computes τ^* as follows:

$$\tau^* = \frac{1}{m} \sum_{j=1,\ldots,m} (\psi(q_j, \beta^*)). \quad (24)$$

In other words the difference between the two methods is that M2 averages thresholds that were "adapted" to β^*, while M1 averages the original thresholds.

Our preliminary experiments demonstrated that M2 does not offer advantage over M1 while requiring somewhat greater

computational effort. Therefore, we only present results of using M1.

5.4.3 Global Fusion

Each of the methods of Section 5.4.1 can be applied to the combined set of document scores and relevance judgments across the training queries. This is the global approach used in [1]. We compare results obtained with this approach to those of our federated method for each of the local methods L1, L2, logistic and centroid.

6. RESULTS

Our experiments involved 9-fold cross-validation over the 99 queries. Each combination was trained on 8 sets of 11 queries and evaluated on the remaining 11-query set.

Tables 1-4 contain results for the 4 possible combinations of the three individual classifiers. Each table contains results for both federated and global combination, and for individual systems. Since the scores of the individual systems do not depend on the combination method used, we show them only once for each table. The column named "Max" contains the score that would have been obtained if somehow the best of the single systems for a particular query could have been guessed in advance and applied to that query. The tables also show the percentage of queries for which combinations were better than the individual systems or the Max.

Several conclusions can be drawn from these results. In general, the Bayesian-kNN and Bayesian-Rocchio combinations showed much better performance in comparison to the individual methods than the Rocchio-kNN combinations, or even the triple combination Bayesian-kNN-Rocchio. In retrospect, it is clear to us that kNN and Rocchio models that we used behave similarly. When training individual classifiers, we chose the settings for each that showed the best performance. This resulted in a kNN method with a large $k = 384$, making kNN rather similar to the Rocchio method. Thus, our results support the intuition that combination works best when different (independent) systems are combined.

We found that the federated approach, i.e. learning combination weights over one set of queries and applying it to another *can* improve the performance. This is however difficult. An improvement in average utility score over individual classifiers was achieved only using the federated approach, not the global approach, and only with the centroid method for determining the weights. These results are given in bold font in Tables 1 and 2.

It appears that while combining classifiers on a single query is advantageous, attempts to carry over the combination to other queries lead to using parameters that are not optimal. The trade-off between information gain resulting from combining classifiers and less-than-optimal coefficients resulting from combining coefficients over queries determines the extent of improvement (or loss) of performance.

Of the four methods for computing combination weights, the centroid method is clearly the best for the federated approach. It is not so clear which method is the best for the global approach. The federated approach outperformed the global method in all but 3 combinations. The 3 exceptions were Bayesian-kNN logistic, Bayesian-Rocchio logistic and Bayesian-Rocchio-kNN logistic combinations. In all of these three cases the logistic method was used for computing local weights. Also, we note that L2 method gives better results than L1 method in 5 out of 8 combinations (with both the federated and the global approaches) and is worse only in 2 combinations.

Head-to-head comparisons of untruncated utility T11NU (in Table 5) confirms the results presented in Tables 1-4 and the conclusions drawn from comparing T11SU scores.

However, our results look quite different if we use a different measure of success. For each combination we computed the percentage of queries where combination performed better than the original systems. Looking at Tables 1 and 2, we notice that the combination methods are outperforming the individual classifiers on more than half the queries, and are beating Max on a large fraction of the queries. Also, the global approach is showing better performance than the federated approach.

Finally, Table 6 directly compares the federated approach to the global approach based on the percentage of queries on which the federated approach performs better than the global approach, and on which it performs worse. The comparison is clearly in favor of the global approach, especially for the Bayesian-Rocchio and Bayesian-kNN combinations, which resulted in good utility scores.

Taking into consideration the fact that federated approach results in better truncated utility, and looking at the actual query scores (not given here), we conclude that the federated approach does better on the queries where the global approach does badly, but performs comparably or slightly worse on the remaining queries.

We believe we understand the reasons for such behavior. The difficult queries are usually the ones for which few relevant documents are available in the training set. When we are computing weights for the global approach, a few relevant documents for a query do not significantly affect the weights, especially when many queries with a lot of relevant documents are present. Therefore, the resulting weights are going to do well on queries where there were a lot of relevant documents, while on the queries with very few relevant documents the results will be poor. The federated method averages the optimal weights for each query. Therefore, a query with few relevant documents has as much influence on the final set of weights as does a query with a lot of relevant documents. As a result, the federated approach produces the set of weights that does not perform as well as that of a global approach on the queries with a lot of training documents, but does better on the queries with few relevant training documents.

These observations suggest that it may be beneficial to develop methods for estimating when a combination or a particular single classifier performs well on a given query, for example based on the number of relevant documents available for training. While some work in this direction has been done ([5], [2]), new results and advances are to be expected.

7. CONCLUSION

Our results demonstrate the difficulty of constructing a query-independent combination that can beat good individual systems, especially when these systems are not independent, as in the case with our Rocchio and kNN methods.

We have examined four methods of computing coefficients, two of which have not been previously seen in the literature. One of the new methods, centroid ("centroid") gave the best results, in terms of both the truncated and the untruncated measures, over all combination methods. In 2 of the 4 feder-

	Average T11SU				Percentage of Queries		
	Combination	Bayesian	kNN	Max	> Bayesian	> kNN	> Max
Federated							
L1	0.574	0.578	0.583	0.620	62.6	62.6	45.5
L2	0.575	...	...	...	66.7	67.7	50.5
logistic	0.549	...	...	...	36.4	48.5	22.2
centroid	**0.587**	...	...	...	55.6	68.7	42.4
Global							
L1	0.569	...	...	...	69.7	65.7	53.5
L2	0.569	...	...	...	67.7	66.7	50.5
logistic	0.556	...	...	...	53.5	59.6	43.4
centroid	0.569	...	...	...	62.6	62.6	47.5

Table 1: Bayesian-kNN Combination Results.

	Average T11SU				Percentage of Queries		
	Combination	Bayesian	Rocchio	Max	> Bayesian	> Rocchio	> Max
Federated							
L1	0.563	0.578	0.540	0.606	55.6	69.7	44.4
L2	0.573	...	...	...	63.6	71.7	49.5
logistic	0.542	...	...	...	37.4	57.6	27.3
centroid	**0.583**	...	...	...	58.6	73.7	46.5
Global							
L1	0.562	...	...	...	63.6	68.7	52.5
L2	0.566	...	...	...	69.7	68.7	56.6
logistic	0.577	...	...	...	61.6	74.7	51.5
centroid	0.572	...	...	...	67.7	73.7	57.6

Table 2: Bayesian-Rocchio Combination Results.

	Average T11SU				Percentage of Queries		
	Combination	Rocchio	kNN	Max	> Rocchio	> kNN	> Max
Federated							
L1	0.475	0.540	0.583	0.597	34.3	14.1	11.1
L2	0.479	...	...	...	31.3	15.2	12.1
logistic	0.494	...	...	...	39.4	18.2	12.1
centroid	0.496	...	...	...	40.4	17.2	13.1
Global							
L1	0.433	...	...	...	31.3	13.1	09.1
L2	0.441	...	...	...	35.4	20.2	15.2
logistic	0.430	...	...	...	40.4	25.3	22.2
centroid	0.439	...	...	...	38.4	19.2	18.2

Table 3: Rocchio-kNN Combination Results.

	Average T11SU					Percentage of Queries			
	Combination	Bayesian	kNN	Rocchio	Max	> Bayesian	> Rocchio	> kNN	> Max
Federated									
L1	0.579	0.578	0.583	0.540	0.627	64.6	68.7	72.7	44.4
L2	0.574	...	...	...	...	66.7	67.7	72.7	46.5
logistic	0.552	...	...	...	...	36.4	39.4	66.7	18.2
centroid	0.581	...	...	...	...	51.5	62.6	75.8	31.3
Global									
L1	0.570	...	...	...	...	70.7	65.7	71.7	49.5
L2	0.569	...	...	...	...	68.7	66.7	70.7	45.5
logistic	0.571	...	...	...	...	56.6	65.7	72.7	37.4
centroid	0.568	...	...	...	...	63.6	62.6	76.8	44.4

Table 4: Bayesian-kNN-Rocchio Combination Results.

Method	Bayesian-kNN		Bayesian-Rocchio		Rocchio-kNN		Bayesian-kNN-Rocchio	
	federated	global	federated	global	federated	global	federated	global
L1	0.111	-0.011	0.032	-0.099	-0.324	-1.606	0.036	-0.055
L2	0.025	-0.053	0.137	-0.081	-0.234	-1.435	0.022	-0.041
logistic	-0.043	-0.121	-0.144	0.086	-0.261	-1.585	0.001	-0.033
centroid	0.124	-0.125	0.185	0.016	-0.272	-1.394	0.141	-0.187

Table 5: Average T11NU for both federated and global approaches.

Method	Bayesian-kNN		Bayesian-Rocchio		Rocchio-kNN		Bayesian-kNN-Rocchio	
	>	<	>	<	>	<	>	<
L1	0.354	0.444	0.182	0.444	0.444	0.444	0.273	0.455
L2	0.303	0.444	0.242	0.374	0.404	0.474	0.283	0.455
logistic	0.222	0.576	0.152	0.697	0.455	0.455	0.212	0.646
centroid	0.343	0.535	0.242	0.485	0.465	0.444	0.333	0.576

Table 6: Percentage of queries for which federated combination performs better than global combination (">") and those where it is performs worse ("<") (based on T11SU).

ated combinations it lead to a better performance than any of the individual classifiers.

We also have compared the global and federated approaches. The federated approach appears to do better on difficult queries, while the global approach does somewhat better on the remaining queries. However, the federated approach produces results with better utility scores. We suggest an explanation for this phenomenon that has to do with the distribution of the relevant documents across queries and how this affects the combination weights produced by each method. These observations also underscore the importance of a measure (in a more general sense than a particular utility) to the evaluation of a combination of classifiers. In our case, the choice of a "better" method for constructing topic-independent classifier combinations depends on a measure of quality used. Therefore, when constructing or choosing a method for a practical problem, it is most important to choose an appropriate measure of performance.

8. ACKNOWLEDGMENTS

The authors thank the KD-D group for its support through National Science Foundation grant number EIA-0087022 to Rutgers University. The views expressed in this article are those of the authors, and do not necessarily represent the views of the sponsoring agency.

The authors thank Kwong Bor Ng for suggesting centroid approach, Andrei Anghelescu, Vladimir Menkov and Alex Genkin for providing the method scores and results, and all members of the DIMACS Working Group on Monitoring Message Streams (http://www.stat.rutgers.edu/ madigan/mms/) for helpful discussions and suggestions.

9. REFERENCES

[1] B. T. Bartell, G. W. Cottrell, and R. K. Belew. Automatic combination of multiple ranked retrieval systems. In *Proceedings of SIGIR-94*, pages 173–181, 1994.

[2] P. N. Bennett, S. T. Dumais, and E. Horvitz. Probabilistic combination of text classifiers using reliability indicators: Models and results. In *Proceedings of SIGIR-02*, pages 207–215, Tampere, Finland, 2002.

[3] A. Genkin, D. D. Lewis, and D. Madigan. Large-scale bayesian logistic regression for text categorization. In preparation.

[4] D. A. Hull, J. O. Pedersen, and H. Schütze. Method combination for document filtering. In *Proceedings of SIGIR-96*, pages 279–288, Zürich, CH, 1996. ACM Press, New York, US.

[5] W. Lam and K.-Y. Lai. A meta-learning approach for text categorization. In *Proceedings of SIGIR-01*, pages 303–309, New Orleans, US, 2001.

[6] L. S. Larkey and W. B. Croft. Combining classifiers in text categorization. In *Proceedings of SIGIR-96*, pages 289–297, Zürich, CH, 1996. ACM Press, New York, US.

[7] D. D. Lewis, Y. Yang, T. Rose, and F. Li. RCV1: a new benchmark collection for text categorization. *Journal of Machine Learning Research*, 5:361–397, April 2004.

[8] S. Robertson and I. Soboroff. The TREC 2002 filtering track report. In *The Eleventh Text REtrieval Conference (TREC-11)*. NIST, 2002.

[9] Y. Yang, T. Ault, and T. Pierce. Combining multiple learning strategies for effective cross-validation. In *Proceedings of ICML-00*, pages 1167–1182, Stanford, US, 2000. Morgan Kaufmann Publishers, San Francisco, US.