On Feature Distributional Clustering for Text Categorization

Ron Bekkerman
CS Department
Technion
Haifa 32000 Israel
ronb@cs.technion.ac.il

Ran El-Yaniv
CS Department
Technion
Haifa 32000 Israel
rani@cs.technion.ac.il

Naftali Tishby
School of CS and Engineering
and Center for Neural Computation,
The Hebrew University
Jerusalem 91904 Israel
tishby@cs.huji.ac.il

Yoad Winter
CS Department
Technion
Haifa 32000 Israel
winter@cs.technion.ac.il

ABSTRACT

We describe a text categorization approach that is based on a combination of feature distributional clusters with a support vector machine (SVM) classifier. Our feature selection approach employs distributional clustering of words via the recently introduced *information bottleneck method*, which generates a more efficient *word-cluster* representation of documents. Combined with the classification power of an SVM, this method yields high performance text categorization that can outperform other recent methods in terms of categorization accuracy and representation efficiency. Comparing the accuracy of our method with other techniques, we observe significant dependency of the results on the data set. We discuss the potential reasons for this dependency.

1. INTRODUCTION

Text categorization is a fundamental task in information retrieval with rich body of knowledge that has been accumulated in the past 25 years [20]. The "standard" approach to text categorization has so far been using a document representation in a word-based 'input space', i.e. as a vector in some high (or trimmed) dimensional Euclidean space where each dimension corresponds to a word, and then has been relying on some classification algorithm, trained in a supervised learning manner. Since the early days of text categorization, the theory and practice of classifier design has significantly advanced and several strong learning algorithms have emerged (see e.g. [9, 26]). In contrast, despite numerous attempts to introduce more sophisticated document representation techniques, e.g. based on higher order word statistics [17, 1, 22] or NLP [11, 2], the simple minded independent word-based representation, known as *bag-of-words (BOW)*, remained very popular. Indeed, to-date the best multi-class, multi-labeled categorization results for the well-known Reuters-21578 data set [3] are based on the BOW representations [10, 13].

In this paper we give further evidence to the usefulness of a more sophisticated text representation method, which is based on applications of the recently introduced *Information Bottleneck (IB)* clustering framework [17, 24, 1, 22]. Specifically, in this approach, IB clustering is used for representing a document in a feature *cluster* space (instead of feature space), where each cluster is a distribution over document classes. As we show, this relatively new distributional representation, that was first explored in this context by [1] and then by [22, 23], combined with a Support Vector Machine (SVM) classifier [26, 7], allows for high performance categorization of the well known 20 Newsgroups (20NG) data set [16]. In contrast, we show that the categorization of 20NG using the strong algorithmic word-based setup of Dumais *et al.* [10], which achieved the best reported categorization results for the Reuters data set, is inferior.

At the outset, these findings are perhaps not surprising since the use of distributional word clusters (instead of words) for representing documents, has several advantages. First, the word clustering performs a sophisticated dimensionality reduction, which implicitly considers correlations between the various features (terms or words). In contrast, popular greedy approaches for feature selection such as mutual information, information gain and TFIDF (see [27]) only consider each feature individually. Second, the clustering achieved by the IB method provides a good solution to the statistical sparseness problem that is prominent in the straightforward word-based document representation. Finally, the clustering of words allows for extremely compact representations (with minor information compromises) that allow the use of strong but computationally intensive classifiers.

Nevertheless, when we tested our categorization setup (with word cluster representation) on the Reuters data set (ModApte split) we could not obtain any accuracy improvement over the best known categorization results of Dumais *et al.* (word-based representation). We hypothesize that this difference occurs because the articles in the Reuters data set were categorized on the basis of only a few *keywords*. If this hypothesis is correct it might mean that with respect to this data set, no significant improvement can be achieved by representations that are more sophisticated than bag-of-words.

SIGIR'01, September 9-12, New Orleans, Louisiana, USA.

In Section 5 we present our study of this question and our attempts to characterize the differences between the 20NG and the Reuters data sets.

The rest of this paper is organized as follows. In Section 2 we discuss known categorization results for the two data sets we consider (20NG and Reuters) and previous work that investigated the use of word cluster representation for text. In Section 3 we briefly present the algorithmic components we use starting from mutual information for feature selection, the information bottleneck method and distributional clustering, the deterministic annealing clustering algorithm and support vector machines. Although each of these components has been known and used, we believe this is the first attempt to apply all these components together. In Section 4 we present our experimental setup and in Section 5 we give a detailed description of our results. Finally, in Section 6 we summarize our conclusions.

2. RELATED RESULTS

Dumais *et al.* [10] reported on the best-known multi-label categorization of the Reuters data set (ModApte split). Dumais *et al.*'s method applies the Support Vector Machines (SVM) learning scheme over a reduced BOW representation, where the feature reduction is based on a greedy word mutual information to the class labels. This method leads to a break-even result of 92.0% on the 10 largest categories of Reuters. Joachims [13] uses an SVM for a multi-labeled categorization of the Reuters data set as well, but without feature reduction, and achieves break-even of 86.4%. In [12] Joachims also investigate *uni-labeled* categorization of the 20NG data set and using the Rocchio algorithm of [18] applied on a mutual information reduced BOW representation, he obtains 90.3% accuracy.

Using the distributional clustering scheme of Pereira *et al.* [17], Baker and McCallum [1] apply a distributional clustering of words, represented as distributions over their classes (that is, classes of documents in which they appear), to generate a more sophisticated text representation via word clusters. In [1], this representation is applied to the 20NG data set, using a Naive Bayes classifier over the word clusters. The result is 85.7% accuracy, using a *uni-labeled* categorization. Baker and McCallum also compared their methods to other feature reduction techniques such as clustering words with Latent Semantic Indexing (see e.g. [8]), mutual information [27] and Markov "blankets" feature selection [14] (the classifier was Naive Bayes in all cases). Their conclusion is that categorization based on word-cluster representation is slightly inferior in accuracy to categorization based on BOW but the word-cluster representation is significantly more efficient.

In this paper we investigate the strength of the word clustering approach for document representation. This type of distributional clustering is essentially a supervised application of the *Information Bottleneck (IB)* method of Tishby *et al.* [24]. In [22], Slonim and Tishby explore the properties of this word cluster representation and motivate it within the more general IB method. Finally, in [23], the same authors show that categorization with representation based on IB-clustering of words can actually improve the categorization accuracy compared to BOW representation whenever the training set is small. These results are obtained using a Naive Bayes classifier and the data sets are a number of class subsets of the 20NG.

3. METHODS AND ALGORITHMS

3.1 Feature selection via mutual information

Feature selection (or feature reduction) is a general term for techniques for dimensionality reduction. Considering a (high dimensional) vectorial representation of the data, these techniques attempt to select an optimal subset of vector components onto which data points will be projected. The incentives are to improve classification quality (via noise reduction) and to reduce computational complexity. The selection of an optimal feature subset is a hard problem that suffers from a combinatorial explosion. Therefore, despite the existence of some sophisticated methods (see e.g. [14]) many authors consider simple and greedy approaches [27]. Dumais *et al.* [10] used the following method, based on *mutual information (MI)*. Let c and w be binary random variables indicating whether or not the category c and the word w occurred. The mutual information between c and w is defined as follows:

$$I(w,c) = \sum_{e_w \in \{0,1\}} \sum_{e_c \in \{0,1\}} P(e_w, e_c) \log \frac{P(e_w, e_c)}{P(e_w)P(e_c)} \quad (1)$$

where e_w and e_c are boolean indicator random variables of the word w and the category c, respectively. In one of the experimental setups described below we use this mutual information technique for feature selection.

3.2 Information bottleneck and distributional clustering

Distributional clustering using mutual information optimization was introduced by Pereira, Tishby, and Lee [17] for distributions of verb-object pairs. The original algorithm aimed at minimizing the average distributional similarity (in terms of the Kullback-Leibler divergence [5]) between the conditional $P(verb|noun)$ and the noun centroids distributions. This algorithm turned out to be a special case of a more general principle, termed *The Information Bottleneck (IB) Method* by Tishby, Pereira, and Bialek [24]. Here the question of relevant encoding of one variable with respect to another variable was posed and formulated, and a general converging algorithm introduced.

Relevant encoding of the random variable X relies on (soft) partitioning of X into domains that preserve the mutual information between X and another given variable, Y. The resulting partition, or clusters of X, constitute an approximate *sufficient partition* that enables the construction of an optimal code (e.g. binary tree) over X, that provides all the information that X has on Y. Denoting the induced partition, or set of clusters, by $\tilde{X}$, the problem has a simple variational formulation: *maximize the mutual information $I(\tilde{X}, Y)$ with respect to the partition $P(\tilde{X}|X)$, under a constraint on $I(\tilde{X}, X)$*. Namely, find the optimal tradeoff between the minimal partition of X and the maximum preserved information on Y.

The resulting self consistent equations essentially coincides with the original distributional clustering algorithm and can be written as

$$P(\tilde{X}|X) = \frac{P(\tilde{X})}{Z(\beta, X)} \exp\left[-\beta \sum_{Y} P(Y|X) \ln\left(\frac{P(Y|X)}{P(Y|\tilde{X})}\right)\right],$$

where $Z(\beta, X)$ is a normalization factor, and $P(Y|\tilde{X})$ in the exponential is defined implicitly, through Bayes' rule, in terms of the partition (assignment) rules $P(\tilde{X}|X)$, $P(Y|\tilde{X}) = \frac{1}{P(\tilde{X})}\sum_X P(Y|X)P(\tilde{X}|X)P(X)$. The parameter β is a Lagrange multiplier introduced for the constrained information. Viewed as an inverse temperature it can be used as an *annealing* parameter to choose a desired resolution.

3.3 Distributional clustering via deterministic annealing

The IB self-consistent equations can be iterated and are guaranteed to converge for every value of β. This is in fact analogous to the Blahut-Arimoto algorithm in information theory [5]. The value of β can be modified, from very low (high "temperature") which corresponds to very poor distributional resolution, to very high (low "temperature") which corresponds to higher resolution (i.e. more clusters). This procedure, known as *deterministic annealing*, was introduced in the context of clustering by Rose et. al. [19]. We employed this top-down hierarchical clustering procedure in this work. When applying this algorithm one has to use an appropriate annealing rate in order to identify "phase transitions" which correspond to cluster splits. Note that for small data sets an alternative agglomerative algorithm that avoids this problem has been developed in [22] (see also the approximate faster agglomerative procedure in [1]).

3.4 Support Vector Machines (SVMs)

The *support vector machine (SVM)* [25, 7] is an inductive learning scheme that has recently proved to be successful along various application domains. In particular, some evidences indicate that SVM is also a good choice for text categorization [12, 10]. Whenever the data is linearly separable, linear SVM computes the maximum margin linear classifier. For the non-linearly separable case there is an extension [4] that allows for cost dependent training errors. Several authors advocated the choice of linear SVM (as opposed to kernel-based SVM) due to their speed in both training and classification time and their generalization abilities with respect to textual domains. In all our experiments we used a linear SVM. The implementation we used was the SVM*light* package of Joachims [15]. In the multi-labeled setting (see below) we applied standard binary "threshold" SVMs and in the uni-labeled setting we applied confidence-rated (binary) SVMs that generate instance distances to decision boundaries.

3.5 Putting it all together

A straightforward approach to dealing with multi-class, multi-labeled categorization with m classes is to decompose the problem into m binary problems. There exist recent decomposition methods that seem to be more powerful (see e.g. [6]). Nevertheless, for simplicity and for comparison with related results we chose this straightforward decomposition.

In the case of uni-labled (multi-class) categorization we again used the above decomposition into m binary problems and employed a standard "max-win" approach whereby a document is categorized into a class whose classifier has the maximum confidence rate among all the classifiers.

We present and experiment with two algorithmic schemes. The first one is based on BOW using the greedy mutual information feature selection technique (see Equation (1)) whereby the k most discriminating words (words) are selected, the articles are projected on them and then the SVM classifier is trained on the projected articles (for details see Algorithm 1). The second scheme is based on IB distributional clustering whereby the words of the training set documents are clustered into k clusters ("pseudo-words") using the deterministic annealing implementation of the information bottleneck method (see Sections 3.3 and 3.2, respectively), and the rest of the procedure is similar to the first setup except that articles are now projected onto pseudo-words and not on words (for details see Algorithm 2). Both the algorithmic schemes described are for the multi-labeled setup. The uni-labeled variants can be obtained straightforwardly.

BOW-based training
Input: $C = (c_1, ..., c_m)$ - set of categories
$D_{train} = (d_1, ..., d_n)$ - document training set,
$d_i = \langle B_i, C_i\rangle$ where B_i is a BOW representation of d_i and C_i is the set of categories d_i belongs to
k - feature set size
Output: $H = (h_1, ..., h_m)$ - set of binary classifiers
$(W_1, ..., W_m)$ - set of selected features for each category

Let W_{train} be the set of words in D_{train}
for each category $c_i \in C$ **do**
  Initiate $T_i^+ \leftarrow \emptyset$ set of positive examples
  Initiate $T_i^- \leftarrow \emptyset$ set of negative examples
  for each word $w \in W_{train}$ **compute** $I(w, c_i)$ according to Eq (1)
  Sort words in W_{train} according to $I(w, c_i)$
  Extract k top words $W_i \leftarrow (w_1, ..., w_k)$
  for each article $d = \langle B_j, C_j\rangle \in D_{train}$ **do**
    Project d on W_i: $Projection(d) \leftarrow \langle B_j \cap W_i, C_j\rangle$
    if $c_i \in C_j$ **then**
      Add $Projection(d)$ to T_i^+
    else
      Add $Projection(d)$ to T_i^-
    end if
  end for
  Train an SVM over T_i^+ and T_i^- to construct a binary classifier h_i
end for

BOW- based classification
Input: $d = \langle B_j, C_j\rangle$ - a test article
$H = (h_1, ..., h_m)$ - set of binary classifiers
$(W_1, ..., W_m)$ - set of selected features for each category
Output: $L = (l_1, ..., l_m)$ - set of boolean labels, where $l_i \in \{0, 1\}$ (1 means that d belongs to c_i)

for each classifier $h_i \in H$ **do**
  Project d on W_i: $Projection(d) \leftarrow \langle B_j \cap W_i, C_j\rangle$
  Run h_i on $Projection(d)$ to obtain l_i
end for

Algorithm 1: Multi-labeled MI feature selection + SVM

4. EXPERIMENTAL SETUP

4.1 The data sets

The Reuters-21578 corpus contains 21578 articles taken from the Reuters newswire [3]. Each article is typically designated to one or more semantic categories such as "earn",

IB-based training

Input: $C = (c_1, ..., c_m)$ - set of categories
$D_{train} = (d_1, ..., d_n)$ - document training set,
$d_i = \langle B_i, C_i \rangle$ where B_i is a BOW representation of d_i and C_i is the set of categories d_i belongs to
k - feature set size

Output: $H = (h_1, ..., h_m)$ - set of binary classifiers
f - the mapping function of words to pseudo-words

```
Let W_train be the set of words in D_train
for each word w in W_train do
    Build a vector
    v_w ← (N_w(c_1), ..., N_w(c_m)) / Σ_{c_j} N_w(c_j),
    where N_w(c_i) is number of w-occurrences in c_i
end for
Cluster the set of vectors v_w onto k clusters PW =
(pw_1, pw_2, ..., pw_k) using the IB method
for each word w in W_train do
    Map the w to its closest pseudo-word pw_i: set f(w) =
    pw_i
end for
for each article d = ⟨B_j, C_j⟩ in D_train do
    Project d on PW: Projection(d) ← ⟨f(B_j), C_j⟩
end for
for each category c_i in C do
    Initiatialize T_i^+ ← Ø; T_i^- ← Ø
    for each article d ∈ D_train do
        if c_i ∈ C_j then
            Add Projection(d) to T_i^+
        else
            Add Projection(d) to T_i^-
        end if
    end for
    Train an SVM over T_i^+ and T_i^- to construct a binary
    classifier h_i
end for
```

IB-based classification

Input: $d = \langle B_j, C_j \rangle$ - a test article
$H = (h_1, ..., h_m)$ - set of binary classifiers
f - mapping of words into word-clusters

Output: $L = (l_1, ..., l_m)$ - set of boolean labels, where $l_i \in \{0, 1\}$ (1 means that d belongs to c_i)

```
for each classifier h_i ∈ H do
    Project d on PW: Projection(d) ← ⟨f(B_j), C_j⟩
    Run h_i on Projection(d) to obtain l_i
end for
```

Algorithm 2: Multi-labeled IB word clustering + SVM

"trade", "corn" etc. and the total number of categories is 118. We used the ModApte split, which consists of a training set containing 7063 articles and a test set containing 2742 articles.[1] In both the training and test sets we preprocessed each article so that any additional information except for the title and the body was removed.

The 20 Newsgroups (20NG) corpus contains 19997 articles taken from the Usenet newsgroups collection [16]. Each article is designated to one or more semantic categories and the total number of categories is 20, all of them are of about the same size. Most of the articles have only one semantic tag and about 4.5% of the articles have two or more labels. Following [21] we used the "Xrefs" field of the article headers to detect multi-labeled documents and to remove duplications. We preprocessed each article so that any additional information except for the subject and the body was removed.[2] In addition, we filtered out lines which seemed to be a part of binary files sent as attachments or text delimiters. A line is considered to be a "binary" (or a delimiter) if it is longer than 50 symbols and contains no blanks. Overall we removed 23057 such lines (where most of these occurrences appeared in a small number of articles). In both data sets we lowered the case of all letters and following [10] in the Reuters set we generated distinct features for all the words appearing article titles.

4.2 Cross-validated training and parameter setting

Since the standard split of Reuters is fixed, cross-validation is not applicable. In our experiments with the 20NG data set we used 4-fold cross-validation. That is, we split it randomly and uniformly into 4 parts, roughly 5000 articles in each part (250 articles in each category). In each random partition we used 3/4 of the articles for training and the remaining 1/4 for testing. Note that this split to 3/4 and 1/4 is proportional to the training to test set size ratios in the ModApte split of Reuters.

In order to improve the results we tuned the SVM algorithm parameters. We used linear SVM*light* and tuned the parameters C (which controls training error costs) and J (cost-factor for negative and positive examples). For both parameters we fixed a set of reasonable values and then tested the SVM classifier using all possible combinations with respect to a validation subset, which following [10], was selected to be 1/3 random subset of the training set.

Parameter setting is different for multi-labeled and uni-labeled categorization. For the multi-labelled setting we set the parameters of each individual (binary) classifier independent of the other classifiers. The processes of parameter tuning in our uni-labeled setting is more complex. Since we use the max-win approach, the categorization of a document is dependent on all the binary classifiers involved. For instance, if all the classifiers except for one are perfect, this last bad classifier can generate maximum confidence rates for all the documents, which results in extremely poor performance. Therefore, a global tuning of all the binary classifiers is necessary. However, in the case of the 20NG, where we have 20 binary classifiers, a global exhaustive search is too time consuming and a clever search in this high dimensional parameter space can be considered. Instead we simply utilized the information we have on the 20NG to reduce the size of the parameter space. Specifically, among the 20 categories of 20NG there are some highly correlated ones and we split the list of the categories into 9 groups as in Table 1. For each group the parameters were tuned together and independently from other groups.

[1] Note that in these figures we count documents with at least one label. The original split contains 9603 training documents and 3299 test documents where the additional articles have no labels.

[2] Note that the subject and body of each article were generated by the article's author and the rest of the information in the header (except for the designated newsgroups, which are class labels) is generated by mail routers and servers and can contain predictive information about the class lables.

talk.religion.misc soc.religion.christian alt.atheism
sci.med
sci.electronics comp.sys.mac.hardware comp.sys.ibm.pc.hardware
sci.crypt
talk.politics.guns talk.politics.mideast talk.politics.misc
comp.os.ms-windows.misc comp.graphics comp.windows.x
rec.autos rec.motorcycles
rec.sport.hockey rec.sport.baseball
sci.space

Table 1: A split of 20NG's categories into thematic groups.

4.3 Performance measures

When measuring the performance of a multi-class multi-labeled categorization it is meaningless to use the standard *accuracy* measure. It has been customary to use instead either a *break-even point (BEP)*, which (for a binary categorization problem) is the arithmetic average of *precision* and *recall*, or the *F-measure* which is the harmonic average of them.[3] When considering a categorization into m classes $c_1, \ldots, c_m$, we use a binary decomposition to m classifiers $h_1, \ldots, h_m$, where the i-th classifier is responsible for discriminating between c_i and the rest of the classes. For each classifier h_i we compute a confusion matrix of four entries $\alpha_i, \beta_i, \gamma_i$ and δ_i where α_i counts the number of samples that were classified by h_i into category c_i whose true label sets include c_i; β_i counts the number of samples that were classified by h_i into c_i but their label sets do not include c_i; similarly, γ_i (and δ_i, respectively) count the number of samples that were classified $\neg c_i$ by h_i and their true label sets do (respectively, do not) contain c_i. Thus, the precision of h_i equals $\frac{\alpha_i}{\alpha_i+\beta_i}$ and its recall is $\frac{\alpha_i}{\alpha_i+\gamma_i}$. The total ('micro-averaged') precision P and recall R are given by:

$$P = \frac{\sum_i \alpha_i}{\sum_i \alpha_i + \sum_i \beta_i} \qquad R = \frac{\sum_i \alpha_i}{\sum_i \alpha_i + \sum_i \gamma_i}.$$

Finally, the total micro-averaged BEP is given by $\frac{P+R}{2}$ and the total micro-averaged F-measure is $\frac{1}{1/P+1/R}$. Note that the micro-averaged precision and recall are simply weighted averages (weighted by class sizes) of the precisions and recalls of the individual classifiers. Following Dumais *et al.* (and for comparison with this work) we report the total micro-averaged BEP in all the multi-labeled categorization experiments below.

In the uni-labeled experiments we used the traditional *accuracy* measure. since we used the 20NG (multi-labeled) set also for the uni-labeled experiment, we followed Joachims [12] and considered a (uni-labeled) categorization of a document to be correct if the labeled assigned by the (max-win) classifier was among the set of this document's possible labels (note that the data set we used is multi-labeled).

5. RESULTS AND DISCUSSION

5.1 Multi-labeled categorization

[3]The break-even measure may favor trivial results; for example, if no data were categorized properly, then the recall is zero and precision is 1, so their average is 0.5 instead of 0 when using the harmonic average.

Table 2 summarizes the multi-labeled categorization results obtained by the above two methods over the Reuters (10 largest categories) and the 20NG data sets. Note that the 92.0% result for the Reuters data set was established by Dumais *et al.* in [10].

Our results show an interesting difference in the quality of the two methods described above, when applied to the Reuters and 20NG data sets. First, as far as we know, the 88.6% BEP is the first reported result for a multi-labeled categorization of the 20NG data set. Previous attempts for multi-labeled categorization of this set were performed by [21] (without overall reported performance). When we computed the (micro-averaged) BEP corresponding to the BOW representation (following the setting described in [10]) we could not obtain results better than 79.8% even when we "unfairly" allowed the algorithm to tune its parameters over the respective test sets (for each of the folds). This result (which is obtained of course under unrealistic conditions), can indicate an upper bound on the accuracy that can be achieved using this algorithmic setup. However, when we increased the number of features (selected using MI) up to 15000, the results improved (using the "unfair" parameter tuning) to 86.3%. Thus, we respect to 20NG, the algorithmic setup based on distributional word-clusters outperforms the BOW setup with respect to both categorization accuracy (BEP) and representation efficiency.

We repeated the same experiment over the Reuters data set but here we obtained opposite results. Now the IB word-cluster representation lost its advantage and achieved a BEP of 91.2% (note that under the "unfair" parameter tuning the IB-based algorithm achieved a BEP of 92.5%), which is slightly smaller than the results obtained using the BOW representation as reported by [10].[4]

Algorithmic Setup	BEP (Reuters)	BEP (20NG)
SVM + MI selection ($k = 300$)	92.0 [10]	79.8 ± 0.5 (unfair)
SVM + MI selection ($k = 15000$)	92.0	86.3 ± 0.5 (unfair)
SVM + IB clustering ($k = 300$)	91.2	88.6 ± 0.3

Table 2: Multi-labeled categorization BEP results for 20NG and Reuters. k is the number of selected words or word-clusters. All 20NG results are averages of 4-fold cross-validation. 'Unfair' indicates unfair parameter tuning over the test sets

What makes the performance of these two representation methods different over these data sets? Why does the BOW representation outperforms the IB-based representation over Reuters?

Perhaps the key to the answer is related to the process which generated the *labeling* of these data sets. As noted by Lewis (see [3]), the Reuters-21578 set contains articles that appeared on the Reuters newswire in 1987 and were assembled and indexed into categories by a few personnel from Reuters Ltd. Presumably, the manual indexing of the

[4]We could not reproduce the 92.0% BEP result reported by [10] ($k = 300$) and the best (fair) result we achieved for this case was 91.1%.

Reuters articles relied mainly on a restricted set of keywords that the indexers looked for. In contrast, the articles in the 20NG were labeled by their own creators (by choosing and sending the articles to the recipient newsgroups), and their annotation relied on full understanding of the articles and their context.

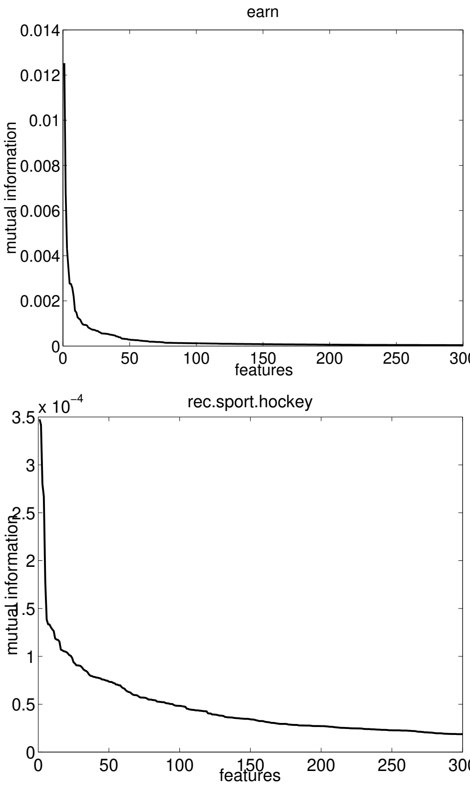

Figure 1: Mutual Information sorted histograms of best discriminating features for two categories. Top: earn of Reuters; Bottom: rec.sport.hockey of 20NG

In order to test this hypothesis, for each category in both data sets we computed the mutual information between words appearing in the category and the category. Then we sorted these words by decreasing values of their mutual information. For instance, in Figure 1 we show two graphs of the MI behavior and it could be seen that the graph of "earn" (Reuters) goes down much sharper than the one of rec.sport.hockey (20NG), which indicates that only a few words of Reuters contribute to the text categorization. As can be seen, the scales of the y-axis of the two graphs differ by one order of magnitude. In order to compare them we plot them in Figure 2 on a percentage scale where each mutual information value is linearly transformed so that a value of x in a dynamic range of $[a, b]$ is transformed to $(x - a)/(b - a)$. When we consider the dynamic range of the 300 most informative words in each category we obtain the normalized (and sorted) histograms of Figure 2. When put on the same scale, the graphs indicate that the 20NG categories are characterized by more features than those of Reuters.

In Figure 3 we show two learning curves plotting the obtained BEP as a function of the number of words used. In

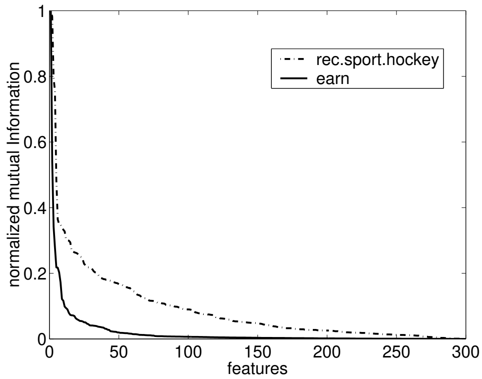

Figure 2: Normalized Mutual Information sorted histograms of both 'earn' (Reuters) and 'rec.sport.hockey' (20NG)

the figure we see two curves: one, which describes the learning rate with respect to Reuters, and the second with respect to 20NG. As can be seen, the BEP of Reuters almost approaches its maximum with only 50 words (that were chosen with the greedy, non-optimal mutual information method). This means that other words do not contribute much. However, the graph of 20NG constantly goes up while its slope constantly lowers.

In addition, we note that with only one word per category the BEP result for the entire Reuters corpus is 74.6% while for 20NG it is much lower (40.7%). Finally, in Table 3 we list the individual BEP result for categorizing the 10 largest categories in Reuters based on three words. For instance, based on the words "vs", "cts" and "loss" it is possible to achieve a BEP categorization of *earn* which is over 93%. We note that the word "vs" appears in 87% articles of category *earn* (that is, 914 articles among total 1044 in this category). This word appears in only 15 non-*earn* articles in the test set and therefore "vs" can, by itself, categorize *earn* with very high precision.[5] This phenomenon was already noticed by Joachims [12] who showed that a classifier built on only one word ("wheat") can lead to extremely high accuracy of distinguishing between the category *wheat* and the others within a uni-labeled setting.

5.2 Uni-labeled categorization

We performed similar categorization experiments on 20NG using the uni-labeled setting. The results are shown in Table 4. As it is in the multi-labeled setting, the advantage of word-clusters (the IB scheme) over BOW is clear when using the same (300) number of features. However, if we increase the number of selected words this gap decreases and with $k = 15000$ words the BOW accuracy result is essentially the same (note however that the reported BOW results were obtained using unfair parameter tuning). In contrast, when we increase the number of word *clusters* up to 1000 we do not achieve any significant accuracy gain (see Figure 4).

Thus, in the uni-labeled setting, the accuracy disadvantage of the BOW representation can be better traded-off us-

[5]On the training set "vs" appears in 1900 of the 2709 earn articles (70.1%) and only in 14 of the 4354 non-*earn* articles (0.3%)

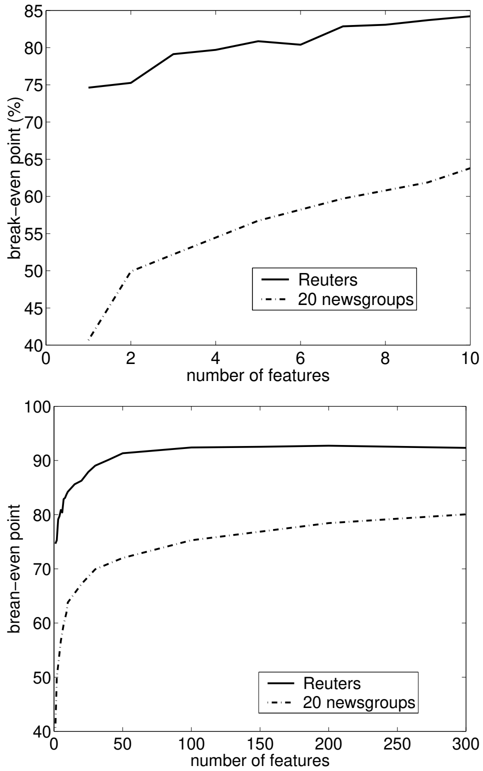

Figure 3: Learning curves (BEP vs. number of words) for the Reuters-21578 and the 20NG over the Mutual Information top 10 (above) and the top 300 (below) words using BOW-based representation and SVM

ing more words. Unlike to the multi-labeled setting where we observed both accuracy and representation efficiency advantage of the word-clusters representation, here we only observe representation efficiency advantage over BOW.

5.3 Computational efforts

We ran our experiments on a Pentium III 600MHz 2G RAM PC under Windows 2000. For the setup of the MI feature selection and SVM classification, the computational bottleneck was the SVM training, for which a single run could take a few hours, depending on the parameter values. In general, the smaller the parameters C and J are, the quicker the algorithm runs.[6]

As for IB-clustering and SVM classification, the SVM*light* runs faster on the input vectors of pseudo-words. However, the clustering itself can take up to one hour on the entire 20NG set, and requires much memory (up to 1G RAM for a run). The overall training and test time over the entire 20NG in the multi-labled setting is about 16 hours (4 hours for each of the 4 cross-validation folds).

The computational bottleneck in the uni-labeled setting on 20NG is the SVM parameter tuning. The IB-based experiment runs for about 45 hours, while the word-based experiment runs for a little bit less than 96 hours (4 days).

[6]All the C values we tested were < 1; we failed to run the SVM*light* on our 20NG partitions with parameter values ≥ 1.

Category	1st word	2nd word	3rd word	BEP
earn	vs+	cts+	loss+	93.5%
acq	shares+	vs−	Inc+	76.3%
money-fx	dollar+	vs−	exchange+	53.8%
grain	wheat+	tonnes+	grain+	77.8%
crude	oil+	bpd+	OPEC+	73.2%
trade	trade+	vs−	cts−	67.1%
interest	rates+	rate+	vs−	57.0%
ship	ships+	vs−	strike+	64.1%
wheat	wheat+	tonnes+	WHEAT+	87.8%
corn	corn+	tonnes+	vs−	70.3%

Table 3: Three best words in terms of MI and their categorization BEP rate of the 10 largest categories of Reuters. The micro-average over these categories is 79.1%. '+' means that the word contributes by its appearance, '−' means that the word contributes by its disappearance

Algorithmic Setup	Accuracy (20NG)
SVM + MI selection ($k = 300$)	85.5 ± 0.7 (unfair)
SVM + MI selection ($k = 15000$)	90.9 ± 0.2 (unfair)
SVM + IB clustering ($k = 300$)	91.0 ± 0.3

Table 4: Uni-labeled categorization accuracy of 20NG. k is the number of selected words or word-clusters. All accuracies are averages of 4-fold cross-validation

6. CONCLUDING REMARKS

We have shown that a cluster-based representation of texts using the Information Bottleneck method, combined with a Support Vector Machine classifier, leads to both uni- and multi-labeled categorization of the 20NG data set that is superior to the best known word-based techniques. We believe that these results show the potential advantage of more sophisticated text representations. On the 20NG data set, with respect to multi-labeled categorization, we obtain both accuracy (BEP) and representation efficiency advantages. With respect to uni-labeled categorization we gain no accuracy advantage but still maintain representation efficiency.

Nevertheless, we found no advantage to our technique in the (multi-labeled) categorization of the Reuters set and we hypothesize that this is due to some inherent differences in the ways the two data sets were generated. Clearly, future work in text categorization could benefit from a comparative study with respect to a larger variety of data sets.

7. ACKNOWLEDGEMENTS

This research was supported by the Israeli Ministry of Science. We sincerely thank Thorsten Joachims and Rob Schapire for their generous help in preparation of this paper. We would also like to thank Susan Dumais, Andrew McCallum, Yoram Singer, Noam Slonim and Tong Zhang for their prompt responses and fruitful discussions. R. El-Yaniv is a Marcella S. Gelltman Academic Lecturer.

8. REFERENCES

[1] L. D. Baker and A. K. McCallum, *Distributional*

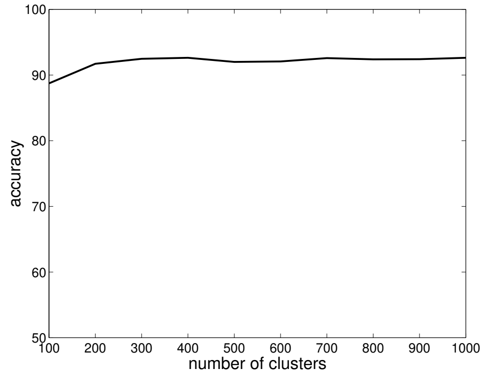

Figure 4: Accuracy vs. number of word-clusters; uni-labeled setting, 20NG

clustering of words for text classification, Proceedings of SIGIR'98, 1998.

[2] Roberto Basili, Alessandro Moschitti, and Maria T. Pazienza, *Language-sensitive text classification*, Proceedings of RIAO-00, 6th International Conference "Recherche d'Information Assistee par Ordinateur" (Paris, France), 2000, pp. 331–343.

[3] The Reuters-21578 collection can be achieved at: http://www.research.att.com/ lewis.

[4] C. Cortes and V. Vapnik, *Support vector networks*, Machine Learning 20 (1995), 273–297.

[5] T.M. Cover and J.A. Thomas, *Elements of information theory*, John Wiley & Sons, Inc., 1991.

[6] K. Crammer and Y. Singer, *On the learnability and design of output codes for multiclass problems*, Proceedings of COLT'2000, 2000.

[7] N. Cristianini and J. Shawe-Taylor, *An introduction to support vector machines*, Cambridge University Press, 2000.

[8] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman, *Indexing by latent semantic analysis*, Journal of the American Society for Information Science **41(6)** (1990), 391–407.

[9] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern classification (2nd ed)*, John Wiley & Sons, Inc., New York, 2000.

[10] S. T. Dumais, J. Platt, D. Heckerman, and M. Sahami, *Inductive learning algorithms and representations for text categorization*, Proceedings of ACM-CIKM'98, 1998.

[11] P. S. Jacobs, *Joining statistics with nlp for text categorization*, Proceedings of the Third Conference on Applied Natural Language Processing, 1992, pp. 178–185.

[12] T. Joachims, *A probabilistic analysis of the rocchio algorithm with tfidf for text categorization*, Proceedings of ICML'97, 1997, pp. 143–151.

[13] ——, *Text categorization with support vector machines: Learning with many relevant features*, Proceedings of the Tenth European Conference on Machine Learning, 1998, pp. 137–142.

[14] D. Koller and M. Sahami, *Hierarchically classifying documents using very few words*, Proceedings of ICML'97, 1997, pp. 170–178.

[15] The SVM light software can be achieved at: http://ais.gmd.de/ thorsten.

[16] The 20 newsgroups collection can be achieved at: http://kdd.ics.uci.edu/.

[17] F. Pereira, N. Tishby, and L. Lee, *Distributional clustering of english words*, In Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics, 1993, pp. 183–190.

[18] J. Rocchio, *Relevance feedback in information retrieval*, ch. 14, pp. 313–323, Prentice Hall, Inc., 1971, in The SMART Retrieval System: Experiments in Automatic Document Processing.

[19] K. Rose, *Deterministic annealing for clustering, compression, classification, regression and related optimization problems*, Proceedings of the IEEE **86** (1998), no. 11, 2210–2238.

[20] G. Salton and M. McGill, *Introduction to modern information retrieval*, McGraw Hill, 1983.

[21] R. Schapire and Y. Singer, *Boostexter: A boosting-based system for text categorization*, Machine Learning 39 (2000), 135–168.

[22] N. Slonim and N. Tishby, *Agglomerative information bottleneck*, Advances in Neural Information Processing Systems, 2000, pp. 617–623.

[23] ——, *The power of word clustering for text classification*, Proceedings of the European Colloquium on IR Research, ECIR, 2001.

[24] N. Tishby, F. Pereira, and W. Bialek, *The information bottleneck method*, 1999, Invited paper to The 37th annual Allerton Conference on Communication, Control, and Computing.

[25] V.N. Vapnik, *The nature of statistical learning theory*, Springer-Verlag, 1995.

[26] ——, *Statistical learning theory*, John Wiley & Sons Inc., New York, 1998.

[27] Y. Yang and J.0. Pedersen, *A comparative study on feature selection in text categorization*, Proceedings of ICML'97, 1997, pp. 412–420.