On feature distributional clustering for text categorization

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

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

ABSTRACT

We describe a new powerful text categorization method that is based on a combination of distributional features with a support vector machine (SVM) classifier. Our feature selection approach uses distributional clustering of words via the recently introduced *information bottleneck method*, which generates a more efficient representation of the documents. When combined with the classification power of Support Vector Machines we produce the best known multi-label categorization results on the 20 Newsgroups dataset.

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, 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, first explored in this context by [1, 22, 23], combined with a Support Vector Machine (SVM) classifier [26, 7], allows for the best reported result for a multi-class categorization of another well-known 20 Newsgroups (20NG) dataset [16]. We also show that the categorization of the 20NG using the strong algorithmic word-based setup of Dumais et al. [10], which achieved the best reported categorization results for the Reuters dataset, is significantly 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 striking advantages. First, the word clustering performs a sophisticated dimensionality reduction, which implicitly considers correlations between the various features (terms or words). In contrast, the numerous greedy approaches for feature selection only consider each feature individually (e.g. mutual information, information gain, TFIDF, etc. see [27]). Second, the clustering achieved by the IB method provides a good solution to the statistical sparseness problem common in text categorization when using the representation in feature space. Finally, the clustering of words allows for extremely compact representations (without information compromises) that allow the use of strong classifiers with typically lower computational effort.

However, when we tested our categorization setup (with word cluster representation) on the Reuters dataset (ModApte split) we could not obtain any improvement over the best known categorization results of Dumais et al (word-based representation). We hypothesize that this difference appears because the articles in the Reuters dataset 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. In Section 5 we present our study of this question and our attempts to characterize the differences between the 20NG and the Reuters datasets.

The rest of this paper is organized as follows. In Section 2 we discuss categorization results for the two datasets we consider (20NG and Reuters) and previous attempts to use word cluster representation for text. In Section 3 we present all 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 time all these components have been applied together. In Section 4 we present our experimental setup and 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 dataset (ModApte split). Dumais et al's method is to apply 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. This method leads to a break-even result of 92.0% on the 10 largest categories. Joachims [13] uses an SVM for a multi-label categorization of the Reuters dataset as well, without feature reduction, and achieves break-even of 86.4%.

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, to generate a more sophisticated representation via word clusters. In [1], this representation is applied to the 20NG dataset, 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 was that the word-clustering representation led to the best accuracy.

However, improved results were achieved by Joachims in [12], which as far as we know shows the best results for *uni-labeled* categorization of the 20NG dataset. Joachims applied a Naive Bayes classifier to Rocchio algorithm [18] over a mutual information-based reduced feature representation, which leads to 90.3% accuracy.

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 actually improves the results on BOW representation whenever the training set is small and with respect to a naive Bayes classifier.

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 incentive is to improve classification quality (via noise reduction) or improve performance. 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:

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

where e_w and e_c are boolean random variables of appearance (or disappearance) of the word w and the category c, respectively. In one of the experiment settings described below we used 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 Kulback-Libeler 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 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 enable 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

$$P(\tilde{X}|X) = \frac{P(\tilde{X})}{Z(\beta, X)} \exp\left[-\beta \sum_Y P(Y|X) \ln\left(\frac{P(Y|X)}{P(Y|\tilde{X})}\right)\right], \qquad (2)$$

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 $P(\tilde{X}|X)$,

$$P(Y|\tilde{X}) = \frac{1}{P(\tilde{X})} \sum_X P(Y|X)P(\tilde{X}|X)P(X). \qquad (3)$$

The parameter β is a Lagrange multiplier introduced for the constrained information, and can be used as a natural resolution, or *annealing*, parameter.

3.3 Distributional clustering via deterministic annealing

The above 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 correspond to very poor distributional resolution, to very high (low "temperature") which correspond to higher resolution - more clusters. This procedure, known as *deterministic annealing*, was introduced in the context of clustering by Rose et. al. [19]. We employed the same procedure here, when enabling the increase of the number of clusters during the annealing process (increase of β). The main problem with this procedure is identifying the "phase transitions" that correspond to clusters splits. For small datasets an alternative agglomerative algorithm has been developed by Slonim and Tishby [22], which avoids this problem.

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, there are several pieces of evidence that indicate that SVM is a good choice for text categorization. Following [12, 10] we used the simplest linear SVM. 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 (the basic SVM quadratic optimization problem includes a parameter that controls such training errors costs). 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].

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.

We present two algorithmic setups. The first one is based on feature selection using the mutual information technique (Eq 1). Where the k most discriminating features (words) are selected, the articles are projected on them and then the SVM classifier is trained on the projections (for details see Algorithm 1). The second setup is based on Information Bottleneck Distributional clustering: initially, words of the training set are clustered into k clusters ("pseudo-words") using the deterministic annealing implementation of the information bottleneck method (see 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 best words (Algorithm 2).

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

Let W_{train} be the set of words in D_{train}
for each category $c_i \in C$ **do**
  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
  Run the SVM algorithm on the T_i^+ and T_i^- to construct a binary classifier h_i
end for

Bag of words 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 and 0 means that d does not).

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: 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 designated to zero or more semantic categories such as "earn", "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.

[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.

IB classifier learning
Input: $C = (c_1, ..., c_m)$ - set of categories
$D_{train} = (d_1, ..., d_n)$ - training set of articles, $d_i = \langle B_i, C_i \rangle$ where B_i is a BOW representation of d_i and C_i is a set of categories d belongs to
k - feature reduction size
Output: $H = (h_1, ..., h_m)$ - set of binary classifiers
f - the mapping function of words on pseudo-words

Let W_{train} be the set of words in D_{train}
for each word w in W_{train} **do**
  Build a vector $v_w \leftarrow (N_w(c_1), ..., N_w(c_m))$ where $N_w(c_i)$ is number of occurrences of w in category 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 word w on the appropriate pseudo-word pw_i: $f(w) = pw_i$
end for
for each article $d = \langle B_j, C_j \rangle$ in D_{train} **do**
  Project d on PW: $Projection(d) \leftarrow \langle f(B_j), C_j \rangle$
end for
for each category c_i in C **do**
  for each article $d \in D_{train}$ **do**
    if $c_i \in C_j$ **then**
      Add $Projection(d)$ to T_i^+
    else
      Add $Projection(d)$ to T_i^-
    end if
  end for
  Run the SVM algorithm on the T_i^+ and T_i^- to construct a binary classifier h_i
end for

IB classification
Input: $d = \langle B_j, C_j \rangle$ - a test article
$H = (h_1, ..., h_m)$ - set of binary classifiers
f - the mapping function of words on PW
Output: $L = (l_1, ..., l_m)$ - set of boolean labels, where $l_i \in \{0, 1\}$ (1 means that d belongs to c_i and 0 means that d does not).

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

Algorithm 2: IB word clustering + SVM

The 20 newsgroups 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, however about 7% of them have two and more ones. We extracted the list of categories to which an article belongs from a field "Newsgroups" of the article header. We ignored the problem of duplicated articles.[2] We preprocessed each article so that any additional information except for the subject and the body was removed. In addition, we filtered out lines which seemed to be a part of binary files sent as attachments. A line is considered to be a "binary" if it is longer than 50 symbols and contains no blanks. So that we removed 26 binary attachments and many useless delimiter lines, in a total amount of 23057 lines.

[2]When taking the problem into account, only 4.5% of articles are duplicated, as reported in [21]

4.2 Cross-validated training and parameter setting

Since the standard split of Reuters is fixed, we did not apply cross validation. However, in our experiments with the 20 newsgroups we used 4-fold cross-validation. That is, we split it randomly and uniformly into 4 parts, 4999 articles in each part (250 articles in each category). In each random partition we used 3/4 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 ratios in the ModApte split of Reuters, where the training set is also about 3/4 and the testing set is about 1/4 of the dataset.

In order to improve results we tuned the SVM algorithm parameters. We used the linear SVM setting, so the only parameters we tried to tune were C (trade-off between training error and margin) and J (cost-factor for negative and positive examples). For both parameters we fixed a set of possible values and then we applied the SVM classifier using all their combinations. To perform a fair test, we tuned the parameters on a validation subset which was taken as a random one of the 3 parts of the training set (this corresponds to Dumais's method of tuning parameters described in [10]).

4.3 Performance measure

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* (which is the arithmetic average of *precision* and *recall*) or *F-measure* which is essentially the harmonic average of them.[3] Specifically, when considering a categorization task 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 where their true label sets do (respectively, do not) contain c_i. Thus, that is the precision of h_i equals $\frac{\alpha_i}{\alpha_i+\beta_i}$ and the recall of h_i equals $\frac{\alpha_i}{\alpha_i+\gamma_i}$. 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 micro-averaged break-even point is defined as $\frac{P+R}{2}$ and the micro-averaged F-measure is given by $\frac{1}{1/P+1/R}$.

[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.

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, who used a simple average of the precision and recall (instead of the harmonic average), we also used the simple average of the precision and recall in all the experiments reported here .

4.4 Computational efforts

We ran the tests on the Pentium III 600MHz 2G RAM PC under Windows2000. For the setup of the MI feature selection and SVM classification, the bottleneck was the SVM, 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. For example, we failed to run the SVM*light* on 20NG with a parameter values $C > 1$. However, we managed to improve the run time by filtering the binary attachments out (see 4.1). As for the IB method 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 it requires much memory (up to 1G RAM for a run). The overall training and test time over the entire 20NG is about 28 hours (7 hours for each of the 4 cross-validation folds).

5. RESULTS AND DISCUSSION

Table 1 summarizes the categorization results obtained by the two methods over the Reuters (10 largest categories) and the the 20NG data sets. Note that the 92.0% result for the Reuters data set was established by Dumais et al. in [10]. Another result was obtained by Joachims [12] who did not perform the entire experiment over all the Reuters categories, but made two experiments with two independently chosen categories. He achieved the accuracy of 95.6% on a category "wheat" in a uni-labeled setting.

Our results show an interesting difference in the quality of the two methods described above, when applied to the Reuters and 20NG datasets. First, the break-even of 89.5% is the best reported result for a multi-labeled categorization of the 20NG data set. Previous attempts to categorize this set were performed by [21]. When we computed the micro-averaged break-even point corresponding to the "bare" word representation (following the setting described in [10]) we could not obtain results better than 80.5 ± 0.3 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 serve as an upper bound on the performance of this algorithmic setup.

We repeated the same unfair experiment over the Reuters data set but here we obtained opposite results. Now the IB-based representation lost its advantage, and even under the "unfair" conditions could only achieve a result of 91.6%, which is less successful than the results of the BOW representation.

What makes the performance of these two representation methods so different over these data sets? And why the inferior BOW representation outperformed the IB-based representation?

	Reuters	20NG
SVM + MI selection	92.0 [10]	80.5 ± 0.3 (unfair)
SVM + IB clustering	91.6 (unfair)	89.5 ± 0.3

Table 1: Break-even categorization results for the two data sets, with number of features $k = 300$. The figures which correspond to the 20NG are averages of 4-fold cross validation. 'Unfair' means that the classifier parameters are chosen (unfairly) over the test set - to ensure that any better result cannot be achieved.

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 (Distribution 1.0) 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 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, and their annotation relied on full understanding of the articles and their context.

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 approves the fact that only few words of Reuters contribute maximum 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 to 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 in Figure 2. When put on the same scale, the graphs definitely show that the 20NG categories distinction bases on more features than the one of Reuters.

In Figure 3 we show two learning curves plotting the obtained break-even success rate as a function of the number of words used. In 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 break-even of Reuters approaches its maximum only with 50 words (that were chosen with the greedy, non-optimal mutual information method). This means that other words do not contribute anything. However, the graph of 20NG constantly goes up while its speed of increase constantly lowers.

In addition, we show that with only one word per category the break-even result for the entire Reuters corpus is 74.6% while for 20NG it is much lower (40.7%). In Table 2 we

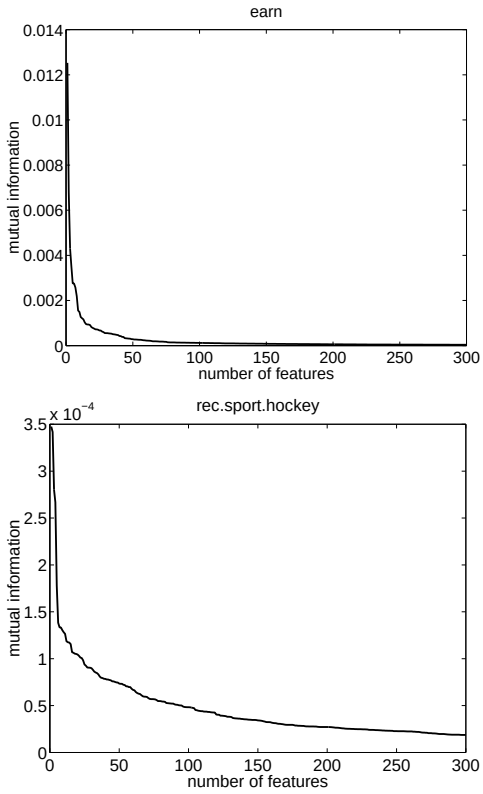

Figure 1: Sorted histograms of best discriminating features for two categories. Above: earn of Reuters; below: rec.sport.hockey of 20NG

list the individual break-even 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 break-even 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.[4] 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 on a uni-labeled setting.

6. CONCLUDING REMARKS AND FUTURE WORK

We have shown that a cluster-based representation of texts using the Information Bottleneck method, combined with a Support Vector Machine classifier, leads to a multi-labeled categorization of the 20NG dataset that is superior to the best known word-based techniques. In our knowledge to-date this result is the best reported multi-labeled break-even on this dataset. We believe that these results show the advantages of more sophisticated text representations than word-based representations, given that they are used in conjunction to a strong classifier like SVM.

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

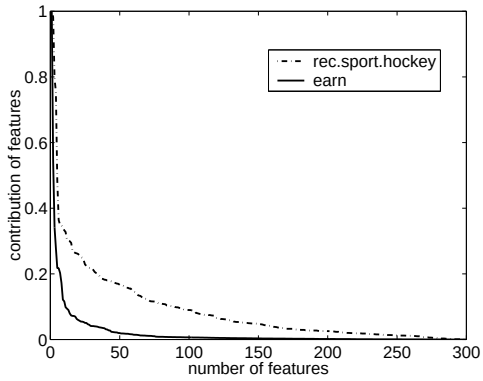

Figure 2: Both 'earn' and 'hockey' on the same scale.

Category	1st word	2nd word	3rd word	Brkeven
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 2: Three best words in terms of MI and their rate of categorization. 10 largest categories of Reuters. The micro-average over these categories is 79.1%. *Plus* means that the word contributes by its appearance, *minus* means that the word contributes by its disappearance

On the other hand, we found no advantage to our technique in the categorization of the Reuters dataset, and we hypothesize that this is due to some inherent differences in the ways the two datasets were generated. This hypothesis should be supported by further research, but we believe that future work in text categorization could benefit from a comparative study of larger variety of datasets.

7. REFERENCES

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

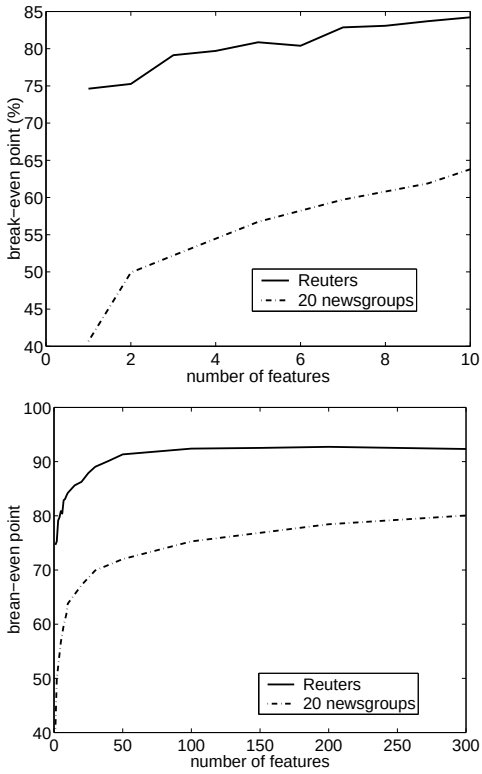

Figure 3: Learning curves (break-even vs. number of words) for the Reuters-21578 and the 20NG over the top 10 (above) and the top 300 (below) words using BOW-based representation and SVM

[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*, To appear in 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.