Investigating Unsupervised Learning for Text Categorization Bootstrapping

Alfio Gliozzo and **Carlo Strapparava**
ITC-irst
Istituto per la Ricerca Scientifica e Tecnologica
I-38050 Trento, Italy
{gliozzo,strappa}@itc.it

Ido Dagan
Computer Science Department
Bar Ilan University
Ramat Gan, Israel
dagan@cs.biu.ac.il

Abstract

We propose a generalized bootstrapping algorithm in which categories are described by relevant seed features. Our method introduces two unsupervised steps that improve the initial categorization step of the bootstrapping scheme: (i) using Latent Semantic space to obtain a generalized similarity measure between instances and features, and (ii) the Gaussian Mixture algorithm, to obtain uniform classification probabilities for unlabeled examples. The algorithm was evaluated on two Text Categorization tasks and obtained state-of-the-art performance using only the category names as initial seeds.

1 Introduction

Supervised classification is the task of assigning category labels, taken from a predefined set of categories (classes), to instances in a data set. Within the classical supervised learning paradigm, the task is approached by providing a learning algorithm with a training data set of manually labeled examples. In practice it is not always easy to apply this schema to NLP tasks. For example supervised systems for Text Categorization (TC) require a large amount of hand labeled texts, while in many applicative cases it is quite difficult to collect the required amounts of hand labeled data. Unlabeled text collections, on the other hand, are in general easily available.

An alternative approach is to provide the necessary supervision by means of sets of “seeds” of intuitively relevant features. Adopting terminology from computability theory, we refer to the standard example-based supervision mode as *Extensional Learning* (EL), as classes are being specified by means of examples of their elements (their *extension*). Feature-based supervision is referred to as *Intensional Learning* (IL), as features may often be perceived as describing the *intension* of a category, such as providing the name or prominent key terms for a category in text categorization.

The IL approach reflects on classical rule-based classification methods, where the user is expected to specify exact classification rules that operate in the feature space. Within the machine learning paradigm, IL has been incorporated as a technique for bootstrapping an extensional learning algorithm, as in (Yarowsky, 1995; Collins and Singer, 1999; Liu et al., 2004). This way the user does not need to specify exact classification rules (and feature weights), but rather perform a somewhat simpler task of specifying few typical seed features for the category. Given the list of seed features, the bootstrapping scheme consists of (i) preliminary unsupervised categorization of the unlabeled data set based on the seed features, and (ii) training an (extensional) supervised classifier using the automatic classification labels of step (i) as the training data (the second step is possibly reiterated, such as by an Expectation-Maximization schema). The core part of IL bootstrapping is step (i), i.e. the initial unsupervised classification of the unlabeled dataset. This step was often approached by relatively simple methods, which are doomed to obtain mediocre quality. Even so, it is hoped that the second step of supervised training would be robust enough to the noise in the initial training set.

The goal of this paper is to investigate additional principled unsupervised mechanisms within the initial classification step, applied to the text categorization. In particular, (a) utilizing a Latent Semantic Space to obtain better similarity assessments between seeds and examples, and (b) applying a Gaussian Mixture (GM) algorithm, which provides a principled unsupervised estimation of classification probability. As shown in our experiments, incorporating these steps consistently improved the accuracy of the initial categorization step, which in turn yielded a better final classifier thanks to the more accurate training set. Most importantly, we obtained comparable or better performance than previous IL methods using *only* the category names as the seed features; other IL methods required collecting a larger number of seed terms, which turns out to be a somewhat tricky task.

Interesting results were revealed when comparing our IL method to a state-of-the-art extensional classifier, trained on manually labeled documents. The EL classifier required 70 (Reuters dataset) or 160 (Newsgroup dataset) documents per category to achieve the same performance that IL obtained using only the category names. These results suggest that IL may provide an appealing cost-effective alternative when sub-optimal accuracy suffices, or when it is too costly or impractical to obtain sufficient labeled training. Optimal combination of extensional and intensional supervision is raised as a challenging topic for future research.

2 Bootstrapping for Text Categorization

The TC task is to assign category labels to documents. In the IL setting, a category C_i is described by providing a set of relevant features, termed an *intensional description* (ID), $id_{c_i} \subseteq V$, where V is the vocabulary. In addition a training corpus $T = \{t_1, t_2, \dots t_n\}$ of *unlabeled* texts is provided. Evaluation is performed on a separate test corpus of labeled documents, to which standard evaluation metrics can be applied.

The approach of categorizing texts based on lists of keywords has been attempted rather rarely in the literature (McCallum and Nigam, 1999; Ko and Seo, 2000; Liu et al., 2004; Ko and Seo, 2004). Several names have been proposed for it – such as *TC by bootstrapping with keywords*, *unsupervised TC*, *TC by labelling words* – where the proposed methods fall (mostly) within the IL settings described here[1].

It is possible to recognize a common structure of these works, based on a typical bootstrap schema (Yarowsky, 1995; Collins and Singer, 1999):

Step 1: *Initial unsupervised categorization.* This step was approached by applying some similarity criterion between the initial category seed and each unlabeled document. Similarity may be determined as a binary criterion, considering each seed keyword as a classification rule (McCallum and Nigam, 1999), or by applying an IR style vector similarity measure. The result of this step is an initial categorization of (a subset of) the unlabeled documents. In (Ko and Seo, 2004) term similarity techniques were exploited to expand the set of seed keywords, in order to improve the quality of the initial categorization.

Step 2: *Train a supervised classifier on the initially categorized set.* The output of Step 1 is exploited to train an (extensional) supervised classifier. Different learning algorithms have been tested, including SVM, Naive Bayes, Nearest Neighbors, and Rocchio. Some works (McCallum and Nigam, 1999; Liu et al., 2004) performed an additional Expectation Maximization algorithm over the training data, but reported rather small incremental improvements that do not seem to justify the additional effort.

(McCallum and Nigam, 1999) reported categorization results close to human agreement on the same task. (Liu et al., 2004) and (Ko and Seo, 2004) contrasted their word-based TC algorithm with the performance of an extensional supervised algorithm, achieving comparable results, while in general somewhat lower. It should be noted that it has been more difficult to define a common evaluation framework for comparing IL algorithms for TC, due to the subjective selection of seed IDs and to the lack of common IL test sets (see Section 4).

3 Incorporating Unsupervised Learning into Bootstrap Schema

In this section we show how the core Step 1 of the IL scheme – the initial categorization – can be boosted by two unsupervised techniques. These techniques

[1] The major exception is the work in (Ko and Seo, 2004), which largely follows the IL scheme but then makes use of labeled data to perform a chi-square based feature selection before starting the bootstrap process. This clearly falls outside the IL setting, making their results incomparable to other IL methods.

fit the IL setting and address major constraints of it. The first is exploiting a generalized similarity metric between category seeds (IDs) and instances, which is defined in a Latent Semantic space. Applying such unsupervised similarity enables to enhance the amount of information that is exploited from each seed feature, aiming to reduce the number of needed seeds. The second technique applies the unsupervised Gaussian Mixture algorithm, which maps similarity scores to a principled classification probability value. This step enables to obtain a uniform scale of classification scores across all categories, which is typically obtained only through calibration over labeled examples in extensional learning.

3.1 Similarity in Latent Semantic Space

As explained above, Step 1 of the IL scheme assesses a degree of "match" between the seed terms and a classified document. It is possible first to follow the intuitively appealing and principled approach of (Liu et al., 2004), in which IDs (category seeds) and instances are represented by vectors in a usual IR-style Vector Space Model (VSM), and similarity is measured by the cosine function:

$$\mathbf{sim}_{vsm}(id_{c_i}, t_j) = \cos(\vec{id}_{c_i}, \vec{t_j}) \qquad (1)$$

where $\vec{id}_{c_i} \in \mathbf{R}^{|V|}$ and $\vec{t_j} \in \mathbf{R}^{|V|}$ are the vectorial representations in the space $\mathbf{R}^{|V|}$ respectively of the category ID id_{c_i} and the instance t_j, and V is the set of all the features (the vocabulary).

However, representing seeds and instances in a standard feature space is severely affected in the IL setting by feature sparseness. In general IDs are composed by short lists of features, possibly just a single feature. Due to data sparseness, most instances do not contain any feature in common with any category's ID, which makes the seeds irrelevant for most instances (documents in the text categorization case). Furthermore, applying direct matching only for a few seed terms is often too crude, as it ignores the identity of the other terms in the document.

The above problems may be reduced by considering some form of similarity in the feature space, as it enables to compare additional document terms with the original seeds. As mentioned in Section 2, (Ko and Seo, 2004) expanded explicitly the original category IDs with more terms, using a concrete query expansion scheme. We preferred using a generalized similarity measure based on representing features and instances a Latent Semantic (LSI) space (Deerwester et al., 1990). The dimensions of the Latent Semantic space are the most explicative principal components of the feature-by-instance matrix that describes the unlabeled data set. In LSI both coherent features (i.e. features that often co-occur in the same instances) and coherent instances (i.e. instances that share coherent features) are represented by similar vectors in the reduced dimensionality space. As a result, a document would be considered similar to a category ID if the seed terms and the document terms tend to co-occur overall in the given corpus.

The Latent Semantic Vectors for IDs and documents were calculated by an empirically effective variation (Gliozzo and Strapparava, 2005) of the *pseudo-document* methodology to fold-in documents, originally suggested in (Berry, 1992). The similarity function $\mathbf{sim}_{lsi}$ is computed by the cosine metric, following formula 1, where $\vec{id}_{c_i}$ and $\vec{t_j}$ are replaced by their Latent Semantic vectors. As will be shown in section 4.2, using such non sparse representation allows to drastically reduce the number of seeds while improving significantly the recall of the initial categorization step.

3.2 The Gaussian Mixture Algorithm and the initial classification step

Once having a similarity function between category IDs and instances, a simple strategy is to base the classification decision (of Step 1) directly on the obtained similarity values (as in (Liu et al., 2004), for example). Typically, IL works adopt in Step 1 a single-label classification approach, and classify each instance (document) to only one category. The chosen category is the one whose ID is most similar to the classified instance amongst all categories, which does not require any threshold tuning over labeled examples. The subsequent training in Step 2 yields a standard EL classifier, which can then be used to assign multiple categories to a document.

Using directly the output of the similarity function for classification is problematic, because the obtained scales of similarity values vary substantially across different categories. The variability in similarity value ranges is caused by variations in the number of seed terms per category and the levels of their generality and ambiguity. As a consequence, choosing the class with the highest absolute similarity value to the instance often leads to selecting a category whose similarity values tend to be generally higher, while another category could have been more similar to the classified instance if normalized similarity values were used.

As a solution we propose using an algorithm based on unsupervised estimation of Gaussian Mixtures (GM), which differentiates relevant and non-relevant category information using statistics from unlabeled instances. We recall that mixture models have been widely used in pattern recognition and statistics to approximate probability distributions. In particular, a well-known nonparametric method for density estimation is the so-called Kernel Method (Silverman, 1986), which approximates an unknow density with a mixture of kernel functions, such as gaussians functions. Under mild regularity conditions of the unknown density function, it can be shown that mixtures of gaussians converge, in a statistical sense, to *any* distribution.

More formally, let $t_i \in T$ be an instance described by a vector of features $\vec{t_i} \in \mathbf{R}^{|V|}$ and let $id_{c_i} \subset V$ be the ID of category C_i; let $\mathbf{sim}(id_{c_i}, t_j) \in \mathbf{R}$ be a similarity function among instances and IDs, with the only expectation that it monotonically increases according to the "closeness" of id_{c_i} and t_j (see Section 3.1).

For each category C_i, GM induces a mapping from the similarity scores between its ID and any instance t_j, $\mathbf{sim}(id_{c_i}, t_j)$, into the probability of C_i given the text t_j, $P(C_i|t_j)$. To achieve this goal GM performs the following operations: (i) it computes the set $\mathcal{S}_i = \{\mathbf{sim}(id_{c_i}, t_j)|t_j \in T\}$ of the similarity scores between the ID id_{c_i} of the category C_i and all the instances t_j in the unlabeled training set T; (ii) it induces from the empirical distribution of values in $\mathcal{S}_i$ a Gaussian Mixture distribution which is composed of two "hypothetic" distributions $\mathcal{C}_i$ and $\overline{\mathcal{C}_i}$, which are assumed to describe respectively the distributions of similarity scores for positive and negative examples; and (iii) it estimates the conditional probability $P(C_i|\mathbf{sim}(id_{c_i}, t_j))$ by applying the Bayes theorem on the distributions $\mathcal{C}_i$ and $\overline{\mathcal{C}_i}$. These steps are explained in more detail below.

The core idea of the algorithm is in step (ii). Since we do not have labeled training examples we can only obtain the set $\mathcal{S}_i$ which includes the similarity scores for all examples together, both positive and negative. We assume, however, that similarity scores that correspond to positive examples are drawn from one distribution, $P(\mathbf{sim}(id_{c_i}, t_j)|C_i)$, while the similarity scores that correspond to negative examples are drawn from another distribution, $P(\mathbf{sim}(id_{c_i}, t_j)|\overline{C_i})$. The observed distribution of similarity values in $\mathcal{S}_i$ is thus assumed to be a mixture of the above two distributions, which are recovered by the GM estimation.

Figure 1 illustrates the mapping induced by GM from the empirical mixture distribution: dotted lines describe the Probability Density Functions (PDFs) estimated by GM for $\mathcal{C}_i$, $\overline{\mathcal{C}_i}$, and their mixture from the empirical distribution ($\mathcal{S}_i$) (in step (ii)). The continuous line is the mapping induced in step (iii) of the algorithm from similarity scores between instances and IDs (x axis) to the probability of the instance to belong to the category (y axis).

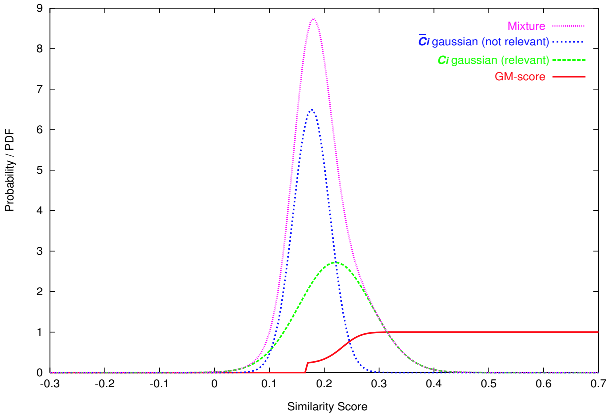

Figure 1: Mapping induced by GM for the category *rec.motorcycles* in the 20newsgroups data set.

The probabilistic mapping estimated in step (iii) for a category C_i given an instance t_j is computed by applying Bayes rule:

$$\begin{aligned} P(C_i|t_j) &= P(C_i|\mathrm{sim}(id_{c_i}, t_j)) = \\ &= \frac{P(\mathrm{sim}(id_{c_i}, t_j)|C_i)P(C_i)}{P(\mathrm{sim}(id_{c_i}, t_j)|C_i)P(C_i) + P(\mathrm{sim}(C_i, t_j)|\overline{C_i})P(\overline{C_i})} \end{aligned} \quad (2)$$

where $P(\mathbf{sim}(id_{c_i}, t_j)|C_i)$ is the value of the PDF of $\mathcal{C}_i$ at the point $\mathbf{sim}(id_{c_i}, t_j)$, $P(\mathbf{sim}(id_{c_i}, t_j)|\overline{C_i})$ is the value of the PDF of $\overline{\mathcal{C}_i}$ at the same point, $P(C_i)$ is the area of the distribution $\mathcal{C}_i$ and $P(\overline{C_i})$ is the area of the distribution $\overline{\mathcal{C}_i}$. The mean and variance parameters of the two distributions $\mathcal{C}_i$ and $\overline{\mathcal{C}_i}$, used to evaluate equation 2, are estimated by the rather simple application of the Expectation Maximization (EM) algorithm for Gaussian Mixtures, as summarized in (Gliozzo et al., 2004).

Finally, following the single-labeled categorization setting of Step 1 in the IL scheme, the most likely category is assigned to each instance, that is, $argmax_{C_i} P(C_i|t_j)$.

3.3 Summary of the Bootstrapping Algorithm

step 1.a: Latent Semantic Space. Instances and Intensional Descriptions of categories (the seeds) are represented by vectors in Latent Semantic space. As an option, the algorithm can work with the classical Vector Space Model using the original feature space. Similarity scores between IDs and instances are computed by the Cosine measure.

step 1.b: GM. The mapping functions $P(C_i|t_j)$ for each category, conditioned on instances t_j, are induced by the GM algorithm. To that end, an Expectation Maximization algorithm estimates the parameters of the two component distributions of the observed mixture, which correspond to the distributions of similarity values for positive and negative examples. As an option, the GM mapping can be avoided.

step 1.c: Categorization. Each instance is classified to the most probable category - $argmax_{C_i} P(C_i|t_j)$.

step 2: Bootstrapping an extensional classifier. An EL classifier (SVM) is trained on the set of labeled instances resulting from step 1.c.

4 Evaluation

4.1 Intensional Text Categorization Datasets

Even though some typical data sets have been used in the TC literature (Sebastiani, 2002), the datasets used for IL learning were not standard. Often there is not sufficient clarity regarding details such as the exact version of the corpus used and the training/test splitting. Furthermore, the choice of categories was often not standard: (Ko and Seo, 2004) omitted 4 categories from the 20-Newsgroup dataset, while (Liu et al., 2004) evaluated their method on 4 separate subsets of the 20-Newsgroups, each containing only 4-5 categories. Such issues make it rather difficult to compare thoroughly different techniques, yet we have conducted several comparisons in Subsection 4.5 below. In the remainder of this Subsection we clearly state the corpora used in our experiments and the pre-processing steps performed on them.

20newsgroups. The 20 Newsgroups data set is a collection of newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups. As suggested in the dataset Web site[2], we used the "bydate" version: the corpus (18941 documents) is sorted by date and divided in advance into a training (60%) set and a chronologically following test set (40%) (so there is no randomness in train/test set selection), it does not include cross-posts (duplicates), and (more importantly) does not include non-textual newsgroup-identifying headers which often help classification (Xref, Newsgroups, Path, Followup-To, Date).

We will first report results using *initial seeds* for the category ID's, which were selected using only the words in the category names, with some trivial transformations (i.e. `cryptography#n` for the category `sci.crypt`, `x-windows#n` for the category `comp.windows.x`). We also tried to avoid "overlapping" seeds, i.e. for the categories `rec.sport.baseball` and `rec.sport.hockey` the seeds are only {`baseball#n`} and {`hockey#n`} respectively and not {`sport#n, baseball#n`} and {`sport#n, hockey#n`}[3].

Reuters-10. We used the top 10 categories (Reuters-10) in the *Reuters-21578* collection Aptè split[4]. The complete Reuters collection includes 12,902 documents for 90 categories, with a fixed splitting between training and test data (70/30%). Both the Aptè and Aptè-10 splits are often used in TC tasks, as surveyed in (Sebastiani, 2002). To obtain the Reuters-10 Aptè split we selected the 10 most frequent categories: `Earn, Acquisition, Money-fx, Grain, Crude, Trade, Interest, Ship, Wheat` and `Corn`. The final data set includes 9296 documents. The initial seeds are only the words appearing in the category names.

[2] The collection is available at `www.ai.mit.edu/people/jrennie/20Newsgroups`.

[3] One could propose as a guideline for seed selection those seeds that maximize their distances in the LSI vector space model. On this perspective the LSI vectors built from {`sport#n, baseball#n`} and {`sport#n, hockey#n`} are closer than the vectors that represent {`baseball#n`} and {`hockey#n`}. It may be noticed that this is a reason for the slight initial performance decrease in the learning curve in Figure 2 below.

[4] available at `http://kdd.ics.uci.edu/databases/-reuters21578/reuters21578.html`).

Pre-processing. In both data sets we tagged the texts for part-of-speech and represented the documents by the frequency of each pos-tagged lemma, considering only nouns, verbs, adjectives, and adverbs. We induced the Latent Semantic Space from the training part[5] and consider the first 400 dimensions.

4.2 The impact of LSI similarity and GM on IL performance

In this section we evaluate the incremental impact of LSI similarity and the GM algorithm on IL performance. When avoiding both techniques the algorithm uses the simple cosine-based method over the original feature space, which can be considered as a baseline (similar to the method of (Liu et al., 2004)). We report first results using only the names of the categories as initial seeds.

Table 1 displays the F1 measure for the 20newsgroups and Reuters data sets, with and without LSI and with and without GM. The performance figures show the incremental benefit of both LSI and GM. In particular, when starting with just initial seeds and do not exploit the LSI similarity mechanism, then the performance is heavily penalized.

As mentioned above, the bootstrapping step of the algorithm (Step 2) exploits the initially classified instances to train a supervised text categorization classifier based on Support Vector Machines. It is worthwhile noting that the increment of performance after bootstrapping is generally higher when GM and LSI are incorporated, thanks to the higher quality of the initial categorization which was used for training.

			Reuters	*20 Newsgroups*
	LSI	*GM*	*F1*	*F1*
	no	no	0.38	0.25
+ bootstrap			0.42	0.28
	no	yes	0.41	0.30
+ bootstrap			0.46	0.34
	yes	no	0.46	0.50
+ bootstrap			**0.47**	**0.53**
	yes	yes	0.58	0.60
+ bootstrap			**0.74**	**0.65**

Table 1: Impact of LSI vector space and GM

4.3 Learning curves for the number of seeds

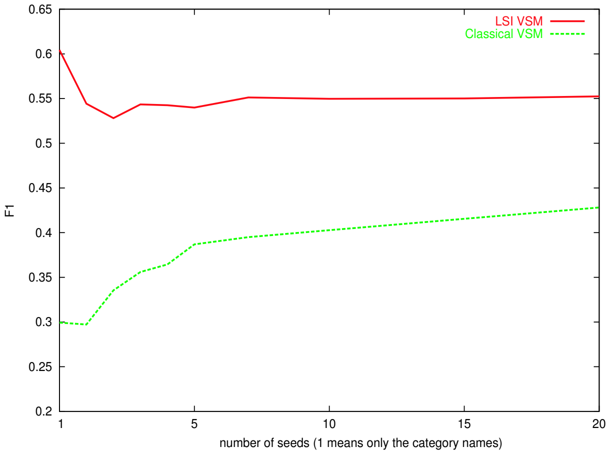

Figure 2: Learning curves on initial seeds for 20 newsgroups, LSI and Classical VSM (no LSI)

This experiment evaluates accuracy change as a function of the number of initial seeds. The experiment was performed for the 20 newsgroups corpus using both the LSI and the Classical vector space model. Additional seeds, beyond the category names, were identified by two lexicographers. For each category, the lexicographers were provided with a list of 100 seeds produced by the LSI similarity function applied to the category name (one list of 100 candidate terms for each category). From these lists the lexicographers selected the words that were judged as significantly related to the respective category, picking a mean of 40 seeds per category.

As seen in Figure 2, the learning curve using LSI vector space model dramatically outperforms the one using classical vector space. As can be expected, when using the original vector space (no generalization) the curve improves quickly with a few more terms. More surprisingly, with LSI similarity the best performance is obtained using the minimal initial seeds of the category names, while adding more seeds degrades performance. This might suggest that category names tend to be highly indicative for the intensional meaning of the category, and therefore adding more terms introduces additional noise. Further research is needed to find out whether other methods for selecting additional seed terms might yield incremental improvements. The current results, though, emphasize the benefit of utilizing LSI and GM. These techniques obtain state-of-the-art performance (see comparisons

[5] From a machine learning point of view, we could run the LSA on the full corpus (i.e. training and test), the LSA being a completely unsupervised technique (i.e. it does not take into account the data annotation). However, from an applicative point of view it is much more sensible to have the LSA built on the training part only. If we run the LSA on the full corpus, the performance figures increase in about 4 points.

in Section 4.5) using only the category names as seeds, allowing us to skip the quite tricky phase of collecting manually a larger number of seeds.

4.4 Extensional vs. Intensional Learning

A major point of comparison between IL and EL is the amount of supervision effort required to obtain a certain level of performance. To this end we trained a supervised classifier based on Support Vector Machines, and draw its learning curves as a function of percentage of the training set size (Figure 3). In the case of 20newsgroups, to achieve the 65% F1 performance of IL the supervised settings requires about 3200 documents (about 160 texts per category), while our IL method requires only the category name. Reuters-10 is an easier corpus, therefore EL achieves rather rapidly a high performance. But even here using just the category name is equal on average to labeling 70 documents per-category (700 in total). These results suggest that IL may provide an appealing cost-effective alternative in practical settings when sub-optimal accuracy suffices, or when it is too costly or impractical to obtain sufficient amounts of labeled training sets.

It should also be stressed that when using the complete labeled training corpus state-of-the-art EL outperforms our best IL performance. This result deviates from the flavor of previous IL literature, which reported almost comparable performance relative to EL. As mentioned earlier, the method of (Ko and Seo, 2004) (as we understand it) utilizes labeled examples for feature selection, and therefore cannot be compared with our strict IL setting. As for the results in (Liu et al., 2004), we conjecture that their comparable performance for IL and EL may not be sufficiently general, for several reasons: the easier classification task (4 subsets of 20-Newsgroups of 4-5 categories each); the use of the usually weaker Naive-Bayes as the EL device; the use of clustering as an aid for selecting the seed terms from the 20-Newsgroup subsets, which might not scale up well when applied to a large number of categories of varying size.

4.5 Comparisons with other algorithms

As mentioned earlier it is not easy to conduct a thorough comparison with other algorithms in the literature. Most IL data sets used for training and evaluation are either not available (McCallum and Nigam, 1999) or are composed by somewhat arbitrary subsets of a standard data set. Another crucial aspect is the particular choice of the seed terms selected to compose an ID, which affects significantly the overall performance of the algorithm.

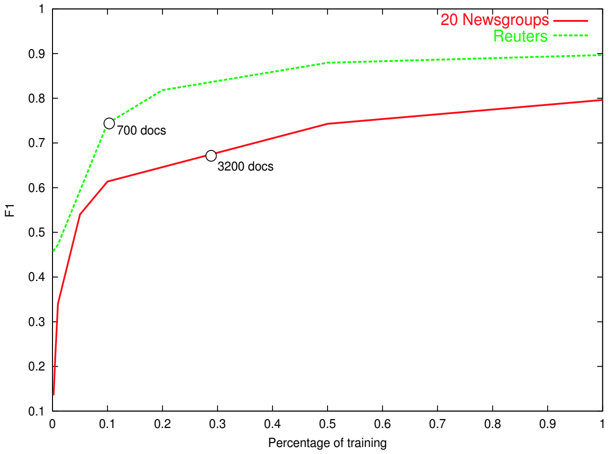

Figure 3: *Extensional* learning curves on as percentage of the training set.

As a baseline system, we implemented a rule based approach in the spirit of (McCallum and Nigam, 1999). It is based on two steps. First, all the documents in the unlabeled training corpus containing at least one word in common with one and only one category ID are assigned to the respective class. Second, a supervised classifier based on SVM is trained on the labeled examples. Finally, the supervised classifier is used to perform the final categorization step on the test corpus. Table 2 reports the F1 measure of our replication of this method, using the category name as seed, which is substantially lower than the performance of the method we presented in this paper.

	Reuters	*20 Newsgroups*
	0.34	0.30
+ *bootstrap*	0.42	0.47

Table 2: Rule-based baseline performance

We also tried to replicate two of the non-standard data sets used in (Liu et al., 2004)[6]. Table 3 displays the performance of our approach in comparison to the results reported in (Liu et al., 2004). Following the evaluation metric adopted in that paper we

[6] We used sequential splitting (70/30) rather than random splitting and did not apply any feature selection. This setting might be somewhat more difficult than the original one.

report here accuracy instead of F1. For each data set (Liu et al., 2004) reported several results varying the number of seed words (from 5 to 30), as well as varying some heuristic thresholds, so in the table we report their best results. Notably, our method obtained comparable accuracy by using just the category name as ID for each class instead of multiple seed terms. This result suggests that our method enables to avoid the somewhat fuzzy process of collecting manually a substantial number of additional seed words.

	Our	*IDs per cat.*	*Liu et al.*	*IDs per cat.*
REC	0.94	1	0.95	5
TALK	0.80	1	0.80	20

Table 3: Accuracy on 4 "REC" and 4 "TALK" newsgroups categories

5 Conclusions

We presented a general bootstrapping algorithm for Intensional Learning. The algorithm can be applied to any categorization problem in which categories are described by initial sets of discriminative features and an unlabeled training data set is provided. Our algorithm utilizes a generalized similarity measure based on Latent Semantic Spaces and a Gaussian Mixture algorithm as a principled method to scale similarity scores into probabilities. Both techniques address inherent limitations of the IL setting, and leverage unsupervised information from an unlabeled corpus.

We applied and evaluated our algorithm on some text categorization tasks and showed the contribution of the two techniques. In particular, we obtain, for the first time, competitive performance using only the category names as initial seeds. This minimal information per category, when exploited by the IL algorithm, is shown to be equivalent to labeling about 70-160 training documents per-category for state of the art extensional learning. Future work is needed to investigate optimal procedures for collecting seed features and to find out whether additional seeds might still contribute to better performance. Furthermore, it may be very interesting to explore optimal combinations of intensional and extensional supervision, provided by the user in the forms of seed features *and* labeled examples.

Acknowledgments

This work was developed under the collaboration ITC-irst/University of Haifa.

References

M. Berry. 1992. Large-scale sparse singular value computations. *International Journal of Supercomputer Applications*, 6(1):13–49.

M. Collins and Y. Singer. 1999. Unsupervised models for named entity classification. In *Proc. of EMNLP99*, College Park, MD, USA.

S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*.

A. Gliozzo and C. Strapparava. 2005. Domains kernels for text categorization. In *Proc. of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, Ann Arbor, June.

A. Gliozzo, C. Strapparava, and I. Dagan. 2004. Unsupervised and supervised exploitation of semantic domains in lexical disambiguation. *Computer Speech and Language*, 18:275–299.

Y. Ko and J. Seo. 2000. Automatic text categorization by unsupervised learning. In *Proc. of COLING'2000*.

Y. Ko and J. Seo. 2004. Learning with unlabeled data for text categorization using bootstrapping abd feature projection techniques. In *Proc. of the ACL-04*, Barcelona, Spain, July.

B. Liu, X. Li, W. S. Lee, and P. S. Yu. 2004. Text classification by labeling words. In *Proc. of AAAI-04*, San Jose, July.

A. McCallum and K. Nigam. 1999. Text classification by bootstrapping with keywords, em and shrinkage. In *ACL99 - Workshop for Unsupervised Learning in Natural Language Processing*.

F. Sebastiani. 2002. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47.

B. W. Silverman. 1986. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall.

D. Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proc. of ACL-95*, pages 189–196, Cambridge, MA.