Text Classification and Co-training from Positive and Unlabeled Examples

François Denis FDENIS@CMI.UNIV-MRS.FR
Anne Laurent ALAURENT@CMI.UNIV-MRS.FR
Équipe BDAA, LIF – UMR 6166, Centre de Mathématiques et d'Informatique (CMI), Université de Provence, 39, rue F. Joliot Curie, 13453 Marseille CEDEX 13, FRANCE

Rémi Gilleron GILLERON@UNIV-LILLE3.FR
Marc Tommasi TOMMASI@UNIV-LILLE3.FR
Équipe Grappa – EA 3588 and projet MOSTRARE – UR INRIA Futurs, Université de Lille 3, domaine universitaire du "Pont de bois", BP 149, 59653 Villeneuve d'Ascq CEDEX, FRANCE

Abstract

In the general framework of semi-supervised learning from labeled and unlabeled data, we consider the specific problem of learning from a pool of positive data, without any negative data but with the help of unlabeled data. We study a naive Bayes algorithm PNB from positive and unlabeled examples. Then, we consider the case where the number of positive examples is quite small, assuming that the co-training setting is relevant, i.e. assuming that the datasets have a natural separation of their features into two sets. We design a co-training algorithm PNCT from positive and unlabeled examples. We give experimental results for the two algorithms PNB and PNCT. They show that text classification with naive Bayes is feasible with positive examples and unlabeled examples and that co-training algorithms can significantly improve learning accuracy when the available set of positive data is small.

1. Introduction

It is often tedious and expensive to hand-label large amount of training data. Thus recently, semi-supervised learning algorithms from a small set of labeled data with the help of unlabeled data have been defined. Such approaches include using Expectation Maximization to estimate maximum a posteriori parameters (Nigam et al., 2000; M.-R & P., 2003), using transductive inference for support vector machines (Joachims, 1999), using the unlabeled data to define a metric or a kernel function (Hofmann, 1999), using a partition of the set of features into two disjoint sets of features (Blum & Mitchell, 1998; Nigam & Ghani, 2000; Muslea et al., 2002).

Here, we consider the problem of learning from positive data with the help of unlabeled data. For instance, in many text learning tasks, such as document retrieval and classification, one goal is the efficient classification and retrieval of interests of some user. Positive information is readily available and unlabeled data can easily be collected. One example is learning to classify web pages as "interesting" for a specific user. Documents pointed by the user's bookmarks define a set of positive examples because they correspond to interesting web pages for him and negative examples are not available at all. Nonetheless, unlabeled examples are easily available on the World Wide Web.

Theoretical results show that in order to learn from positive and unlabeled data, it is sometimes sufficient to consider unlabeled data as negative ones (Denis, 1998; Liu et al., 2002). The starting point of (Liu et al.) is a constrained approximation approach. The idea is to select a function that correctly classifies all positive documents and minimizes the number of unlabeled documents classified as positive. Following this idea, they define a new learning algorithm S-EM built on the naive Bayes classifier in conjunction with the EM (Expectation Maximization) algorithm.

Another approach in the statistical query learning model is to estimate statistical queries over positive and unlabeled examples (Denis et al., 2003). We follow this idea and define a naive Bayes Classifier PNB. PNB takes as input positive and unlabeled data, together with an estimate – possibly rough – of the positive class probability. In practical situations, the positive class probability can be empirically estimated or provided by using some domain knowledge. We compare the performance of S-EM, NB and PNB on three public domain document datasets: the WebKB course dataset, the Reuters collection and the 20 newsgroups dataset.

Next, we consider situations where only a small set of positive data is available together with unlabeled data. In these situations, building accurate classifiers may fail because of the poverty of the input data. However, learning is still possible when the existence of two different views over the data is assumed, as in the co-training framework introduced by Blum and Mitchell (1998). For instance, consider the retrieval of bibliographic references. Positive examples are stored in the user database. A first view consists of the bibliographic fields — title, author, abstract, editor. A second view is the full content of the paper. Unlabeled examples are easily available in the bibliographic databases accessible via the World Wide Web. Co-training algorithms incrementally build basic classifiers over each of the two feature sets. Co-training methods have been used previously to train classifiers in applications like text classification (Blum & Mitchell, 1998), word-sense disambiguation (Yarowski, 1995) and named-entity classification (Collins & Singer, 1999). Co-training learning is a special case of multi-view learning for which semi-supervised learning algorithms have been defined (Muslea et al., 2002).

We define a co-training algorithm PNCT for which the seed information is a small pool of positive documents. At first, PNCT incrementally builds naive Bayes classifiers from positive and unlabeled documents over each of the two views by using PNB. Along the co-training steps, self-labeled positive examples and self-labeled negative examples are added to the training sets. We propose a base algorithm PNNB, which is a variant of PNB, able to use these self-labeled examples. Experiments on the WebKB Course dataset are performed; they show that co-training algorithms lead to significant improvement of classifiers, even when the initial seed is only composed of positive documents.

In Section 2, we present the naive Bayes algorithm from positive and unlabeled data PNB. In Section 3, we define our co-training algorithm PNCT. Experimental results are given in Section 4.

2. Naive Bayes from Positive and Unlabeled Documents

2.1. PNB algorithm

Naive Bayes algorithm from positive and unlabeled examples (PNB) is introduced in (Denis et al., 2002). We briefly present the main ideas of PNB in this section. We only consider binary text classification problems: the set of classes is $\{0,1\}$ where 1 corresponds to the *positive class*. We consider *bag-of-words* representations for documents. Let D be a set of documents and let w be a word. We denote by $N(D)$ the total number of word occurrences in D and by $N(w,D)$ the number of occurrences of w in all the documents of D.

PNB assumes an underlying generative model. In this model, first a class c is selected according to class prior probabilities $P(c)$. Second, a document length l is chosen according to length prior probability $P(l)$. Then, each word w in the document is generated by drawing from a multinomial distribution over words specific to the class $Pr(w|c)$.

The algorithm PNB takes as input: an estimate $\hat{P}(1)$ of the positive class probability $P(1)$, a set PD of positive documents together with a set UD of unlabeled documents. The Positive Naive Bayes classifier PNB classifies a document d consisting of n words $(w_1,\ldots,w_n)$ – with possibly multiple occurrences of a word w – as a member of the class:

$$\mathsf{PNB}(d) = \operatorname*{argmax}_{c\in\{0,1\}} \hat{P}(c) \prod_{i=1}^{i=n} \hat{Pr}(w_i|c)\ . \tag{1}$$

We must now explain how the class probability estimates $\hat{P}(c)$ and the word probability estimates $\hat{Pr}(w_i|c)$ are calculated.

Class probability estimates. An estimate $\hat{P}(1)$ of the positive class probability $P(1)$ is given as input to the learner. An estimate $\hat{P}(0)$ of the negative class probability is set to $1-\hat{P}(1)$.

Positive word probability estimates. We are given as input a set PD of positive documents. We consider the *Laplace smoothing*. The positive word probability estimates are calculated using the following equation:

$$\hat{Pr}(w_i|1) = \frac{1+N(w_i,PD)}{Card(V)+N(PD)} \tag{2}$$

where V is the vocabulary and $Card(V)$ its cardinality.

Negative word probability estimates. When a set ND of negative documents is available, the negative word probability estimates are calculated using the following equation:

$$\hat{Pr}(w_i|0) = \frac{1 + N(w_i, ND)}{Card(V) + N(ND)} \tag{3}$$

But, in our framework negative word probabilities must be estimated without negative examples. Nonetheless, for word probabilities we have:

$$Pr(w_i) = Pr(w_i|0)Pr(0) + Pr(w_i|1)Pr(1) \tag{4}$$

where $Pr(w_i)$ is the probability that the generator creates w_i and $Pr(1)$ is the probability that the generator creates a word in a positive document. Equation 4 can be rewritten as:

$$Pr(w_i|0) = \frac{Pr(w_i) - Pr(w_i|1) \times Pr(1)}{1 - Pr(1)} \tag{5}$$

This equation is used to estimate negative word probabilities. Assuming that the set of unlabeled documents is generated according to the underlying generative model, probability $Pr(w_i)$ is estimated on the set of unlabeled documents by $N(w_i, UD)/N(UD)$. Estimates for negative word probabilities can be rewritten:

$$\hat{Pr}(w_i|0) = \frac{N(w_i, UD) - \hat{Pr}(w_i|1) \times \hat{Pr}(1) \times N(UD)}{(1 - \hat{Pr}(1)) \times N(UD)} \tag{6}$$

In this equation, the positive word probabilities $\hat{Pr}(w_i|1)$ are calculated according to Equation 2 with the input set PD of the positive documents. $\hat{Pr}(1)$ is an estimate of the probability that the generator creates a word in a positive document. As it is assumed that the lengths of documents are independent of the class, $\hat{Pr}(1)$ could be either set to $\hat{P}(1)$ or directly computed using the inputs of PNB (see (Denis et al., 2002) for the calculation and the smoothing of negative word probability estimates).

3. Co-training from Positive and Unlabeled Examples

3.1. Co-training from positive and negative examples

The co-training setting was introduced in (Blum & Mitchell, 1998) in the general framework of learning from labeled data and unlabeled data. The co-training setting applies when a dataset has a natural division of its features. Blum and Mitchell show that under the assumptions that each set of features is sufficient for classification, and the two feature sets of each instance are conditionally independent given the class, PAC-like guarantees on learning from labeled and unlabeled data hold.

They also present a co-training algorithm (see Table 3) which incrementally build naive Bayes classifiers over each of the two views. We denote by D a set of documents described by two views and D_1 (resp. D_2) is the projection of D on the first (resp. second) view. When the documents are labeled, projections are considered together with their labels. The co-training algorithm first creates a pool of u unlabeled documents. It then iterates the following procedure. First, the algorithm trains two classifiers NB_1 and NB_2 based on each of the two views. Second, the classifiers are applied to unlabeled examples. The examples on which the classifiers make the more confident predictions are removed from the set of unlabeled data and are added together with their label to the set of labeled data. At the end, a final hypothesis $Combine(\mathsf{NB}_1, \mathsf{NB}_2)$ is created by a voting scheme that combines the prediction of the classifiers learned in each view.

Following the co-training scheme, we define in Section 3.3 a co-training algorithm from positive and unlabeled examples. A first idea is to replace NB by PNB. Thus along the boosting rounds, only positive and unlabeled examples are used. But, PNB outputs a classifier which can label examples as negative. These self-labeled negative examples should be used along the boosting rounds of the co-training algorithm. With this aim, we first define a variant of PNB which is able to use self-labeled negative examples.

3.2. PNNB algorithm

PNNB takes as input an estimate $\hat{P}(1)$ of the positive class probability and three training sets, a set PD of positive documents, a set ND of negative documents and a set UD of unlabeled documents. The situation differs from classical naive Bayes from labeled examples (the input is a set $D{=}PD \cup ND$ of labeled examples) in two ways:

- the ratio $Card(PD)/(Card(PD) + Card(ND))$ is not an estimate of $P(1)$
- we are not confident in the labels of the negative documents.

As for PNB, the key point is estimating negative word probabilities in Equation 1. The negative word probabilities can be estimated either from the set of negative examples or from the sets of positive and unlabeled examples. We mix these two estimates.

Let us denote by $\hat{Pr}(w_i|0, ND)$ the estimate obtained from the set of negative examples using Equation 3. Let us denote by $\hat{Pr}(w_i|0, PD, UD)$ the estimate obtained from $\hat{P}(1)$ together with the sets PD and UD according to Section 2 by $\hat{Pr}(w_i|0, PD, UD)$. We define estimates for negative word probabilities combining these two estimates using the following equation:

$$\hat{Pr}(w_i|0) = (1-\alpha)\hat{Pr}(w_i|0, PD, UD) + \alpha\hat{Pr}(w_i|0, ND) \tag{7}$$

We set the parameter α to:

$$\alpha = \frac{1}{2} \times \frac{Card(ND)}{Card(PD)} \times \frac{\hat{P}(1)}{1 - \hat{P}(1)} \tag{8}$$

When there is no negative document, α is set to 0 and negative word probabilities are estimated from $\hat{P}(1)$ and the two sets PD and UD according to Equations 7 and 6. When the sets PD and ND are such that the ratio of positive documents in the union set $PD \cup ND$ is equal to the estimate $\hat{P}(1)$ of the positive class probability $P(1)$, α has value 1/2, that is we suppose that we are equally confident on both estimates.

The naive Bayes algorithm PNNB takes as input an estimate $\hat{P}(1)$ of the positive class probability, a set PD of positive documents, a set UD of unlabeled documents and a set ND of negative documents. Class probabilities and positive word probabilities are calculated as for PNB. Negative word probabilities are estimated according to Equations 7 and 8.

3.3. Co-training from only Positive and Unlabeled Examples

We extend the co-training setting to the case where only positive documents and unlabeled documents are given to the learner. The co-training learning algorithm PNCT is given in Table 4. It incrementally builds classifiers over each of the two views with the PNNB algorithm. The co-training process repeats for k iterations. At each co-training step, it picks $Card(PD)/\hat{P}(1)$ documents from the set of unlabeled documents to form the unlabeled dataset given as input to PNNB classifiers. Indeed, large unlabeled datasets can degrade performance of PNB classifiers (Denis et al., 2002). The outcome of the co-training process consists in a final hypothesis whose prediction is obtained by multiplying the prediction of the classifiers learned in each view.

4. Empirical Evaluation

Datasets *The WebKB Course dataset* is a collection of 1051 web pages collected from computer science departments at four universities. Web pages are divided into several categories. We use the student, project, course and faculty categories. No stop-list is used, html tags are removed and no stemming is performed. *The Reuters collection* is the most commonly-used collection for text classification. We use a formatted version of Reuters version 2 (also called Reuters-21450) prepared by Y. Yang and colleagues. Documents are labeled to belong to at least one of the 135 possible categories. Here we consider two binary classification problems defined by the categories *acq* and *grain*. *The 20-newsgroups dataset* contains 20 different UseNet discussion groups. We remove UseNet headers, no stop-list is used and no stemming is performed.

4.1. Experiments with PNB

Preliminary experimental results were given in Denis et al. (2002). They show that PNB is robust against the input value of $\hat{P}(1)$ and compare learning accuracy when varying the number of unlabeled examples. Here, we apply PNB to real world data sets. We also compare PNB and NB considering experimental results for NB as lower and upper bounds for PNB. We also compare PNB and S-EM defined in Liu et al. (2002).

Comparison between PNB and NB Experiments were conducted to compare PNB and NB while varying the number of labeled documents. Results are given in Table 1. For a given row in Table 1, we repeat 200 times the following procedure. We select at random a set of p of labeled data for NB_p, a set of N labeled data for NB_N, and a set of p positive and N unlabeled data for $\text{PNB}_{p,N}$. The Reuters dataset comes with a test set (3662 items) and a train set (9610 items) and we keep this separation in our experiences. In the case of the WebKB dataset, we use as test set the remaining data after drawing the train sets (we obtain therefore 200 different test sets). The estimated error is averaged over the 200 the runs. The standard deviation is estimated as the standard deviation of the accuracy estimations from each holdout run. F score is defined as $F = 2pr/(p + r)$ where p is the precision and r is the recall.

Naive Bayes from positive and unlabeled examples with p positive examples outperforms standard Naive Bayes with p labeled examples. Obviously, if unlabeled documents are given with their correct label, standard Naive Bayes outperforms Naive Bayes from positive and unlabeled examples. Also, it should be noted that we obtain good results when the weight of the positive class is quite small (Category Grain).

Comparison between PNB and S-EM In Table 2, we report error rates and F-measure of PNB and S-EM. Learning algorithms take two sets as input: a set P of positive documents and a set M built with negative documents and positive ones. As indicated by (Liu et al.), the objective is to recover those positive documents put in the mixed M, thus M can be seen as a test set.

The S-EM algorithm takes as input a set P of positive documents, a set M of unlabeled documents and do not need any estimation of $P(1)$. The PNB algorithm takes M as a set of unlabeled documents and P as a set of positive documents. We let $P(1) = 0.5$, considering we have no knowledge about it. Results indicated in Table 2 give the average error rates and F-measure for 100 draws of M and P for PNB. Our algorithm PNB outperforms S-EM in eight of the nine sets of experiments. Results of PNB have also lower variance.

Table 1. A comparison between NB and PNB on three real world datasets.

p	N	NB_p		$\text{PNB}_{p,N}$		NB_N	
		Error	F	Error	F	Error	F
Reuters Category acq; $\hat{P}(1)$ is set to 0.172							
40	232	12(3.3)	66	11(1.8)	67	7.2(1.4)	81
120	698	8.9(2.3)	76	6.9(0.8)	82	4.8(0.5)	88
200	1164	7.4(1.5)	80	5.5(0.5)	86	4.2(0.3)	90
280	1630	6.6(1.1)	83	4.8(0.4)	88	4.0(0.2)	91
360	2096	6.0(0.8)	85	4.5(0.3)	89	3.9(0.3)	91
440	2562	5.7(0.6)	86	4.4(0.3)	90	3.9(0.2)	91
520	3028	5.3(0.5)	87	4.2(0.3)	90	3.9(0.2)	91
Reuters Category grain; $\hat{P}(1)$ is set to 0.045							
40	888	5.2(1.2)	36	3.4(0.6)	61	3.0(0.4)	70
60	1333	4.9(1.1)	40	3.3(0.4)	66	3.0(0.4)	71
80	1777	4.6(0.7)	45	3.3(0.4)	68	3.1(0.4)	72
100	2222	4.3(0.7)	48	3.4(0.4)	69	3.2(0.4)	72
120	2666	4.2(0.6)	51	3.5(0.4)	70	3.3(0.3)	72
140	3111	4.0(0.6)	52	3.6(0.4)	70	3.4(0.3)	71
160	3555	3.9(0.6)	54	3.6(0.3)	70	3.5(0.3)	71
WebKB; $\hat{P}(1)$ is set to 0.22							
10	45	18(7.9)	51	13(4.8)	64	8.3(3.2)	80
20	90	13(5.6)	64	10(3.6)	74	6.3(2.0)	86
30	136	10(4.4)	74	8.9(3.0)	78	5.6(1.7)	87
40	181	8.8(3.5)	79	7.8(2.6)	81	5.0(1.5)	89
50	227	7.8(2.6)	81	7.4(2.2)	82	4.9(1.5)	89
60	272	7.1(2.3)	84	6.9(1.8)	84	4.7(1.4)	89
70	318	7.1(2.8)	83	6.6(1.7)	85	4.5(1.3)	90

Discussion On real world datasets, our algorithm PNB builds accurate classifiers. S-EM and PNB give similar results but PNB needs a rough estimation of $P(1)$. It is worth noting that (Liu et al.) use spy documents in M to optimize the performance of S-EM. Usefulness of spy documents (ten percents of the positive ones in M) is twofold: they are used to avoid strong bias toward positive documents in the EM initialization; they are also used to estimate errors and then select a good classifier in the sequence produced by EM.

4.2. Co-training Experimental Results

We run the PNCT co-training algorithm on the WebKB Course dataset. The binary classification problem is to identify web pages that are course home pages. Each example consists of the words that occur on the web page (*full-text view*), as well as words occurring in the anchor text of hyperlinks pointing to that page (*Hyperlink view*). The class course is designed as the positive class in our setting and 22% of the web pages are positive. Given a fixed seed size $Card(PD)$, for each experiment we first pick at random a test set of 263 documents. From the 819 remaining documents, we draw a set of labeled documents containing $Card(PD)$ positive documents and these positive documents define the seed PD. The remaining documents are left unlabeled and define the set UD. The estimate $\hat{P}(1)$ is set to 0.22. The parameter k is set to the maximal number of co-training steps depending on the number of available unlabeled documents. Parameters p and n are respectively set to 1 and 3.

In a first set of experiments, we study the evolution of error rates along the co-training steps. The seed size $Card(PD)$ is set to 20. Error rates of the output classifiers are averaged over 100 experiments. Figure 1 gives a plot of error versus number of iterations for the PNCT co-training algorithm. Along the first co-training steps, error rates first increase. This may be due to the fact that a sufficient number of self-labeled documents is needed for statistics to become sufficiently accurate. But after some co-training steps, error rates for the full-text classifier and the combined classifier decrease continuously. Finally, the output full-text and combined classifiers outperform classifiers built over the seed dataset. The hyperlink classifier is helped less by co-training but hyperlinks documents contain fewer words. For individual experiments when the seed size is set to 20, we obtain similar plots. When the seed size is lower, for instance consider a seed of 10 positive documents, for some rare experiments, error rates of initial classifiers are quite poor and co-training does not improve the accuracy of the initial classifiers.

We also reproduce experiments of Blum and Mitchell

Table 2. Experimental results from the 20-Newsgroup dataset. Columns PNB and S-EM give accuracy and F measure evaluated on M and averaged over 100 draws (standard deviation is indicated in parenthesis)

Positive	Negative	P	M	pos in M	PNB		S-EM	
					Error	F	Error	F
atheism	rel	200	1400	400	**21.0**(1.4)	58.2(3.5)	27.1(2.8)	61.2(7.4)
graphic	mac	200	1400	400	**10.7**(3.0)	77.9(8.2)	13.6(4.4)	71.9(17.1)
guns	pol	200	1400	400	**14.2**(1.4)	71.8(4.0)	16.2(3.3)	73.7(5.9)
med	elec	200	1400	400	**6.9**(1.5)	86.5(3.4)	8.6(4.8)	81.7(15.8)
oswin	winx	200	1400	400	22.4(3.9)	35.6(18.4)	**20.9**(5.6)	43.0(27.5)
rel	pol	200	1400	400	**13.4**(1.0)	73.9(2.4)	17.0(2.4)	71.6(6.5)
student	course	328	1586	656	**4.1**(0.8)	94.8(1.0)	5.4(1.2)	93.2(1.7)
project	course	100	1132	202	**3.6**(0.7)	90.1(1.9)	6.0(2.8)	79.9(13.4)
faculty	course	224	1378	450	**4.9**(2.2)	92.6(3.2)	7.6(4.3)	86.6(12.8)

(1998) with our implementation of their algorithm (here called CT) of co-training from positive and negative data. As in the PNCT case, along the first co-training steps, error rates first increase and then decreases continuously. We observe that the phenomenon is even accentuated in the CT case. Moreover, it seems to us that the CT algorithm is not robust in the choice of the initial seed. For instance, given 3 positive documents and 9 negative documents, CT ultimately outputs a classifier whose error rate is greater than 12% in 20 percents of our trials. With 10 positive documents in input PNCT ultimately outputs a classifiers whose error rate is greater than 12% in only 4 percents of our trials.

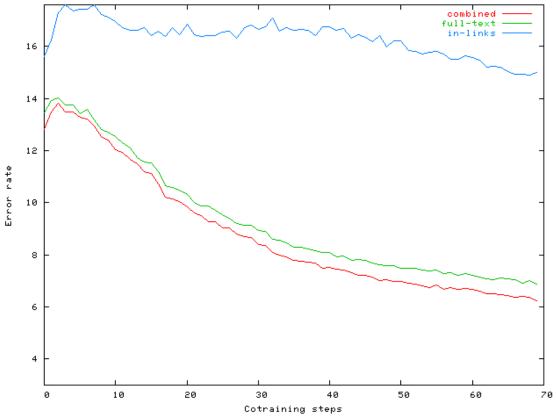

Figure 1. Error versus number of co-training steps for the co-training algorithm PNCT. The seed size is set to 20 and error rates are averaged over 100 experiments

In a second set of experiments, we study error rates of the two co-training algorithms for different seed sizes. For a given seed size, we run 100 experiments. Table 5 gives error rates for the output classifiers. For the co-training algorithm CT defined in (Blum & Mitchell, 1998), we choose an initial seed whose cardinality is close to the number of positive documents in the seed of PNCT in the corresponding row (*e.g.* for the first row of the two tables, 3 positive plus 9 negative documents in the CT seed and 10 positive documents in the PNCT seed). These experimental results on the WebKB dataset are promising. Given a seed of only 10 positive documents and 40 unlabeled documents, the ultimately classifiers produced by PNCT outperform naive Bayes classifiers trained over 90 labeled documents (see Table 1). We should also note that our algorithm seems more robust. For a seed of 20 positive documents, PNCT classifiers always outperform classifiers trained over the seed while for a seed of 12 labeled documents, CT classifiers may be quite poor for some draws of the seed.

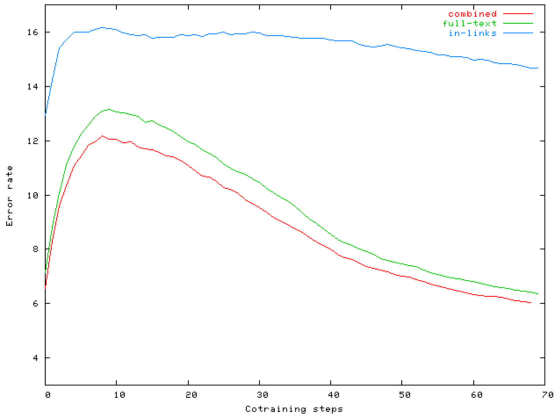

Figure 2. Error versus number of co-training steps for the co-training algorithm CT. The seed size is set to 11+33 documents and error rates are averaged over 100 experiments

Table 3. The co-training algorithm CT (Blum & Mitchell, 1998)

```
Co-training algorithm CT
parameters: $u$, $p$, $n$, $k$
input: a set $D$ of labeled documents; a set $UD$ of unlabeled documents
 Create a pool $UD^p$ choosing $u$ documents at random from $UD$
 Loop for $k$ iterations
  for each $i$ in $\{1, 2\}$
   Use $D_i$ to train a naive Bayes classifier $\mathsf{NB}_i$
   Remove from $UD^p$ the $p$ examples that $\mathsf{NB}_i$ most confidently labels as positive and add them to $D$
   Remove from $UD^p$ the $n$ examples that $\mathsf{NB}_i$ most confidently labels as negative and add them to $D$
  Randomly choose $2n + 2p$ examples from $UD$ to replenish $UD^p$
output: $Combine(\mathsf{NB}_1, \mathsf{NB}_2)$
```

Table 4. The co-training algorithm from positive and unlabeled documents where self-labeled positive and negative documents are added along the co-training steps.

```
Co-training algorithm PNCT
parameters: $p$, $n$, $k$
input: a set $PD$ of positive documents; a set $UD$ of unlabeled documents; an estimate $\hat{P}(1)$
 Set $UD^p$ to $UD$; set $ND$ to $\emptyset$
 Loop for $k$ iterations
  Create a pool $UD^{learn}$ choosing $\frac{Card(PD)}{\hat{P}(1)}$ documents at random from $UD$
  for each $i$ in $\{1, 2\}$
   Train $\mathsf{PNNB}_i$ with input $PD_i$, $UD_i^{learn}$, $ND_i$ and $\hat{P}(1)$
   Remove from $UD^p$ the $p$ examples that $\mathsf{PNNB}_i$ most confidently labels as positive and add them to $PD$
   Remove from $UD^p$ the $p$ examples that $\mathsf{PNNB}_i$ most confidently labels as negative and add them to $ND$
output: $Combine(\mathsf{PNNB}_1, \mathsf{PNNB}_2)$
```

Table 5. Co-training with CT (upper table) and PNCT (below). The column start gives error rate and F-measure of PNB with $Card(PD)$ positive examples and the column stop gives error rate and F-measure for the combined classifiers after k co-training steps.

seed size		steps	Start		Stop	
			Error	F	Error	F
|POS|	|NEG|		CT			
3	9	74	12.4(4.0)	66.8(14.7)	11.4(10.6)	73.6(25.7)
4	12	73	10.4(3.9)	72.9(13.3)	8.7(7.4)	80.3(17.1)
6	18	72	8.4(3.4)	78.1(11.3)	7.87(7)	82.6(16.7)
8	24	71	7.7(2.5)	80.7(8.4)	7.3(4.5)	83.7(11.4)
11	33	69	6.6(2.3)	84.1(6.6)	6.0(1.7)	87.0(3.6)
|POS|	|UNL|		PNCT			
10	40	70	12.8(4.5)	58.4(20.1)	6.3(2.3)	84.9(6.6)
20	80	65	9.6(3.6)	72.0(14.2)	5.1(1.4)	88.2(3.3)
30	120	56	8.2(2.9)	77.8(10.1)	5.0(1.2)	88.5(3.0)
40	160	46	7.1(2.5)	81.3(8.2)	5.1(1.4)	88.4(3.3)
50	200	36	6.5(2.4)	82.9(7.2)	5.0(1.2)	88.4(3.1)

5. Conclusion

We study an adaptation of Naive Bayes that allow to build classifiers from positive and unlabeled data (PNB). The main idea is to approximate word probabilities given the negative class using positive, unlabeled data and an estimation of the weight of the positive class. In the presence of a small set of examples from the target class, we reuse the co-training scheme introduced by Blum and Mitchell (1998) with PNB as a base classifier. We apply these algorithms to a binary text classification problem. Experiments show that starting from a small number of documents from the target class, an estimate of probability of this class and unlabeled documents, our co-training methods build competitive classifiers. Outcomes of the co-training algorithm also seem to be more robust in the choice of the initial seed. Nonetheless, there are still a lot of open questions. How can the positive class probability be estimated from the data? Does an hypothesis testing algorithm apply in our setting ?

Acknowledgements

This research was partially supported by: "CPER 2000-2006, Contrat de Plan état-région Nord/Pas-de-Calais: axe TACT, projet TIC"; fonds européens FEDER "TIC - Fouille Intelligente de données - Traitement Intelligent des Connaissances" OBJ 2-phasing out - 2001/3 - 4.1 - n 3.

We would like to thank the anonymous reviewers for their comments and suggestions.

References

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. *Proc. 11th Annu. Conf. on Comput. Learning Theory* (pp. 92–100). ACM Press, New York, NY.

Collins, M., & Singer, Y. (1999). Unsupervised models for named entity classification. *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora* (pp. 100 – 110).

Denis, F. (1998). PAC learning from positive statistical queries. *Proceedings of the 9th International Conference on Algorithmic Learning Theory (ALT-98)* (pp. 112–126). Berlin.

Denis, F., Gilleron, R., & Letouzey, F. (2003). Learning from positive and unlabeled examples. *Theorical Computer Science*, to appear.

Denis, F., Gilleron, R., & Tommasi, M. (2002). Text classification from positive and unlabeled examples. *IPMU'02, 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems.*

Hofmann, T. (1999). Text categorization with labeled and unlabeled data: A generative model approach. *Working Notes for NIPS 99 Workshop on Using Unlabeled Data for Supervised Learning.*

Joachims, T. (1999). Transductive inference for text classification using support vector machines. *Proceedings of ICML-99, 16th International Conference on Machine Learning* (pp. 200–209).

Liu, B., Lee, W., Yu, P., & Li, X. (2002). Partially supervised classification of text documents. *in Proc. 19th International Conference on Machine Learning* (pp. 387 – 394).

M.-R, A., & P., G. (2003). Semi-supervised learning with explicit misclassification modeling. *Proceedings of the 18th International Joint Conference on Artificial Intelligence.* To appear.

Muslea, I., Minton, S., & Knoblock, C. (2002). Active + Semi-supervised Learning = Robust Multi-view Learning. *Proceedings of ICML-2002* (pp. 435–442).

Nigam, K., & Ghani, R. (2000). Analyzing the applicability and effectiveness of co-training. *Proceedings of CIKM-00, Ninth International Conference on Information and Knowledge Management* (pp. 86–93).

Nigam, K., McCallum, A. K., Thrun, S., & Mitchell, T. M. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning*, *39*, 103–134.

Yarowski, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. *Proceedings thirty-third meeting of the ACL* (pp. 189 – 196).