An HMM-Based Multi-view Co-training Framework for Single-View Text Corpora

Eva Lorenzo Iglesias, Adrían Seara Vieira, and Lourdes Borrajo Diz(✉)

Computer Science Department,
University of Vigo, Escola Superior de Enxeñería Informática, Ourense, Spain
{eva,adrseara,lborrajo}@uvigo.es

Abstract. Multi-view algorithms such as co-training improve the accuracy of text classification because they optimize the functions to exploit different views of the same input data. However, despite being more promising than the single-view approaches, document datasets often have no natural multiple views available.

This study proposes an HMM-based algorithm to generate a new view from a standard text dataset, and a co-training framework where this view generation is applied. Given a dataset and a user classifier model as input, the goal of our framework is to improve the classifier performance by increasing the labelled document pool, taking advantage of the multi-view semi-supervised co-training algorithm.

The novel architecture was tested using two different standard text corpora: Reuters and 20 Newsgroups and a classical SVM classifier. The results obtained are promising, showing a significant increase in the efficiency of the classifier compared to a single-view approach.

Keywords: Hidden Markov Model · Text classification · Co-training · Multi-view

1 Introduction

With the rapid growth of corporate and digital databases, text classification has become one of the key techniques for handling and organizing text data. Supervised text classification algorithms usually require a large number of labelled examples to learn accurately. However, labelling can be a costly and time consuming process because it requires human effort [1,2]. Semi-supervised techniques combine both supervised and unsupervised learning to reduce the need for labelled training data. These algorithms learn a concept definition by combining a small set of labelled examples and a large set of unlabelled ones [2].

Multi-view semi-supervised algorithms are a subset of these techniques. These algorithms take advantage of datasets that have a natural separation of their features or can be described using different "kinds" of information. A prominent example are web pages, which can be classified based on their content as well as on the anchor texts of inbound hyperlinks [3].

F. Martínez-Álvarez et al. (Eds.): HAIS 2016, LNAI 9648, pp. 66–78, 2016.
DOI: 10.1007/978-3-319-32034-2_6

If different views of the dataset are available, then co-training [4] and other multi-view algorithms can improve classification accuracy. The co-training method is a classical algorithm in a multi-view semi-supervised learning technique that trains two independent classifiers, which provide each other with labels for unlabelled data. This algorithm tends to maximize the agreement on the predictions of the two classifiers on the labelled dataset, as well as minimize the disagreement on the predictions on the unlabelled dataset.

In contrast to single-view learning, multi-view learning algorithms introduce one function in order to model a particular view, jointly optimize all the functions to exploit different views of the same input data, and improve the learning performance [4]. However, despite being more promising than single-view approaches, document datasets often have no natural multiple views available, so that only one view may be provided to represent the data.

In this work, we propose an algorithm to generate a new view from a standard text dataset and a co-training framework where this view generation is applied. Given a dataset and a classifier model as input, the goal of our framework is to improve the classifier performance by increasing the labelled document pool, taking advantage of the multi-view semi-supervised co-training algorithm.

The remainder of the manuscript proceeds as follows. The view generation process is described in Sect. 2, and the co-training framework is presented in Sect. 3. In Sects. 4 and 5 we show the experiments and the results obtained for two different text corpora. Finally, the most relevant conclusions are collected at Sect. 6.

2 View Generation

In text classification, given a training set $T = \{(d_0, dl_0), (d_1, dl_1)...(d_n, dl_n)\}$, which consists of a set of pre-classified documents in classes (labels) dl_x, the classifiers are used to model the implicit relation between the characteristics of the document and its class (label), in order to be able to accurately classify new unlabelled documents.

To achieve this, documents need to be expressed in a format that classifying algorithms can handle. The most common approach to represent documents is the Bag-of-Words (BoW) approach [5]. In this case, every document is represented by a vector where elements describe the word frequency (number of occurrences) in that document, as shown in Fig. 1 (a). In addition, each document d_i has an attribute dl_i which has the assigned label as a value.

In order to use a multi-view algorithm like co-training, two representations or views of the documents are needed. One of main goals of this work is to generate a new view from the standard BoW approach in order to apply the co-training algorithm. Specifically, this view generation algorithm is based on Hidden Markov Models (HMMs).

In a previous study, the authors developed an HMM-based document classifier called T-HMM [6]. In this model, HMMs are used to represent sets of documents. An HMM is trained per class (label) with the documents labelled with that label. When a new document needs to be classified, the model evaluates the probability of this document being generated by each of the HMMs, and outputs the label with the maximum probability value.

The goal of the view generation process presented in this paper is to build a new view in which documents are represented by similarities to other groups of documents. Specifically, these document groups are taken from the training set of labelled documents, and a document group is created for each label. This way, each group has all the documents from the training set that share the same label. Every document in the new view is represented by similarities to each document group.

In order to calculate similarities between documents and groups, HMMs are used to represent the groups. One HMM is trained per document group, and the similarity between a document and a group is expressed with the probability of the document being generated by the HMM that represents that group.

Figure 1 shows the complete View Generation process. Firstly, each HMM is trained with a document group. The complete labelled set of the initial dataset represented by the BoW approach is used as the base of the new view. One HMM is created per label, and documents assigned with that label are used to train the HMM.

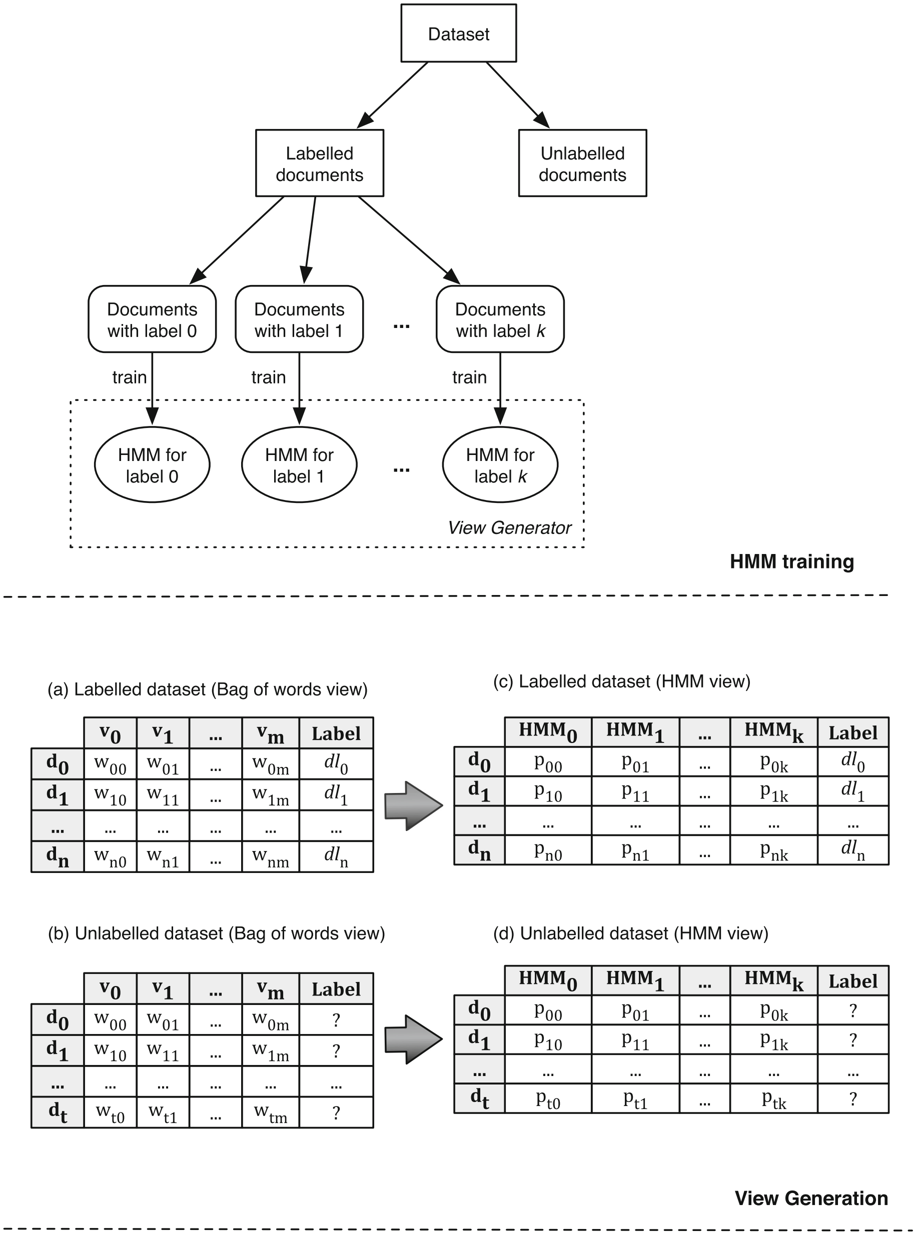
The training process of the HMMs with a document set as input is the same as that described in [6]. HMMs with the same structure as in T-HMM are used to represent each document group. The probability distributions of the HMMs are adjusted automatically depending on the content of the documents, and only two additional parameters need to be fixed to start the process: the number of stats and the generalization factor. Their corresponding values are detailed in the Experiments Section.

Once the HMMs are trained, any document d represented by a BoW approach (labelled or unlabelled) can be also represented in the new HMM view. In order to do so, the probabilities of d being generated by each HMM are calculated using the forward-backward algorithm [6,7]. Finally, the document d in the HMM view is represented by a vector with k elements, where k is the total number of labels, and each element describes the similarity of the document with the document group having the same label represented by the HMM.

Figure 2 shows an example of an HMM view generation. In this case, the documents in the initial document set can be labelled as Relevant (R) or Non-relevant (N). This is a usual scenario in multiple information retrieval systems where a document can be relevant or not to a specific topic.

Using the labelled set of documents represented by a BoW approach as input, one document group is created per label. In this case, the labelled document set is split into Relevant and Non-Relevant document groups. Afterwards, an HMM is trained for each document group using the documents from that group as input: HMM$_R$ and HMM$_N$.

The new HMM view represents any document (labelled or unlabelled) by similarities to the selected document groups. In the example, each document is represented in the new view by similarities to the Relevant and Non-relevant



(a) Labelled dataset (Bag of words view)

	$\mathbf{v_0}$	$\mathbf{v_1}$	...	$\mathbf{v_m}$	**Label**
$\mathbf{d_0}$	w_{00}	w_{01}	...	w_{0m}	dl_0
$\mathbf{d_1}$	w_{10}	w_{11}	...	w_{1m}	dl_1
...	...	...	...	...	...
$\mathbf{d_n}$	w_{n0}	w_{n1}	...	w_{nm}	dl_n

(c) Labelled dataset (HMM view)

	$\mathbf{HMM_0}$	$\mathbf{HMM_1}$	...	$\mathbf{HMM_k}$	**Label**
$\mathbf{d_0}$	p_{00}	p_{01}	...	p_{0k}	dl_0
$\mathbf{d_1}$	p_{10}	p_{11}	...	p_{1k}	dl_1
...	...	...	...	...	...
$\mathbf{d_n}$	p_{n0}	p_{n1}	...	p_{nk}	dl_n

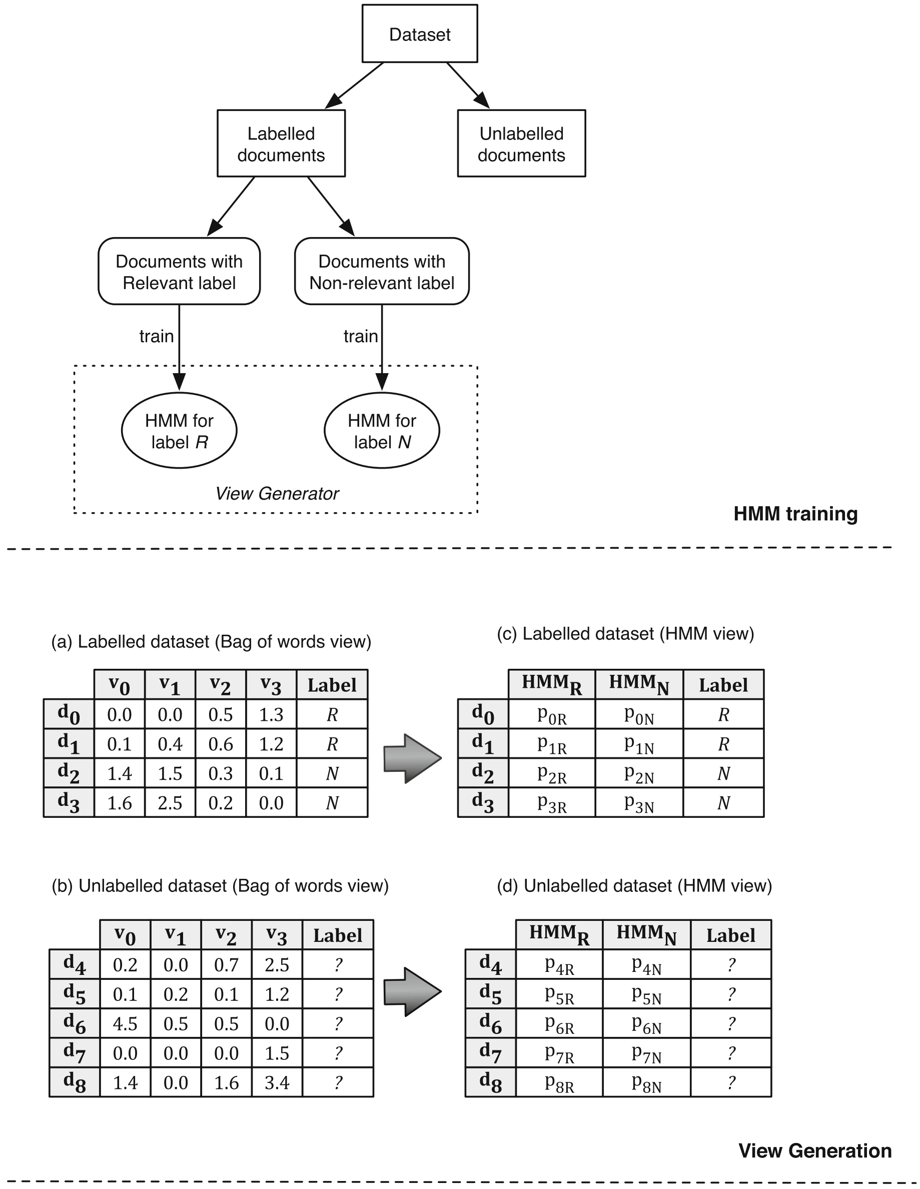
(b) Unlabelled dataset (Bag of words view)

	$\mathbf{v_0}$	$\mathbf{v_1}$	...	$\mathbf{v_m}$	**Label**
$\mathbf{d_0}$	w_{00}	w_{01}	...	w_{0m}	?
$\mathbf{d_1}$	w_{10}	w_{11}	...	w_{1m}	?
...	...	...	...	...	...
$\mathbf{d_t}$	w_{t0}	w_{t1}	...	w_{tm}	?

(d) Unlabelled dataset (HMM view)

	$\mathbf{HMM_0}$	$\mathbf{HMM_1}$	...	$\mathbf{HMM_k}$	**Label**
$\mathbf{d_0}$	p_{00}	p_{01}	...	p_{0k}	?
$\mathbf{d_1}$	p_{10}	p_{11}	...	p_{1k}	?
...	...	...	...	...	...
$\mathbf{d_t}$	p_{t0}	p_{t1}	...	p_{tk}	?

Fig. 1. Creation of the View Generator model and generation of the HMM-view. p_{ij} stands for the probability of the document i being generated by the HMM representing the label j.



(a) Labelled dataset (Bag of words view)

	v_0	v_1	v_2	v_3	Label
d_0	0.0	0.0	0.5	1.3	*R*
d_1	0.1	0.4	0.6	1.2	*R*
d_2	1.4	1.5	0.3	0.1	*N*
d_3	1.6	2.5	0.2	0.0	*N*

(c) Labelled dataset (HMM view)

	HMM_R	HMM_N	Label
d_0	p_{0R}	p_{0N}	*R*
d_1	p_{1R}	p_{1N}	*R*
d_2	p_{2R}	p_{2N}	*N*
d_3	p_{3R}	p_{3N}	*N*

(b) Unlabelled dataset (Bag of words view)

	v_0	v_1	v_2	v_3	Label
d_4	0.2	0.0	0.7	2.5	?
d_5	0.1	0.2	0.1	1.2	?
d_6	4.5	0.5	0.5	0.0	?
d_7	0.0	0.0	0.0	1.5	?
d_8	1.4	0.0	1.6	3.4	?

(d) Unlabelled dataset (HMM view)

	HMM_R	HMM_N	Label
d_4	p_{4R}	p_{4N}	?
d_5	p_{5R}	p_{5N}	?
d_6	p_{6R}	p_{6N}	?
d_7	p_{7R}	p_{7N}	?
d_8	p_{8R}	p_{8N}	?

Fig. 2. Example of the creation of the View Generator model and HMM-view generation. p_{ij} stands for the probability of the document i being generated by the HMM representing the label j

document groups. Since the previously built HMMs act as the representative document group, each document d_i is expressed with a vector containing two elements: the probability of d being generated by HMM$_R$, and the probability of d being generated by HMM$_N$.

It is important to note that the label of the document that needs to be represented in the HMM view is not taken into account in the whole view generation process. Labels are only used in the training phase of the HMMs to create the document groups that share the same label.

3 Co-training with HMM View

In this study, we build a classification framework that, given a single-view dataset and a classifier provided by the user, increases the performance of the classifier by taking advantage of the multi-view classifying process using the previous view generation algorithm.

Specifically, the proposed framework integrates the classifier in a co-training algorithm using both a BoW view and an HMM view of the dataset. In co-training algorithms, one classifier is trained per view. The parameters and the classifier models can be different or the same, but a separate classifier is used for training each view. By maximizing the agreement on the predictions on the unlabelled dataset, the classifiers learn from each other to reach an optimal solution. In each iteration, the classifier on one view labels unlabelled data which are then added to the training pool of both classifiers; therefore, the information is exchanged between the learners [4].

The classic co-training scheme can be seen in Fig. 3. In each iteration, the two classifiers must reach a consensus in the classification of the unlabelled data. Once a set of unlabelled documents is assigned with a label, the documents are added to the training pool of both classifiers to start a new iteration.

Figure 4 shows the proposed co-training scheme. The HMM view is built with the BoW view and both are used in the co-training algorithm. Two instances of the classifier given by the user are created with the same parameters and each one is trained with each view (BoW view and HMM view). The iterative process follows the same structure as the classical co-training algorithm.

To improve the accuracy of the consensus between classifiers, an additional classifier is created. It is important to note that the documents labelled in this stage are included afterwards in the training pool, assuming that the assigned label is correct. This is why the precision of the consensus must be very high, since a misclassification can lead to worse results.

The proposed additional classifier is a distance-based classifier like k-NN. The choice is made based on the capacity of labelling a document with a certain level of confidence. Using a threshold, we can determine that a document is labelled with a certain level of precision. This way, the consensus in each iteration of the co-training algorithm is made between the two base classifiers and the third distance-based classifier using a threshold.

The complete specification of the proposed co-training algorithm is detailed below.

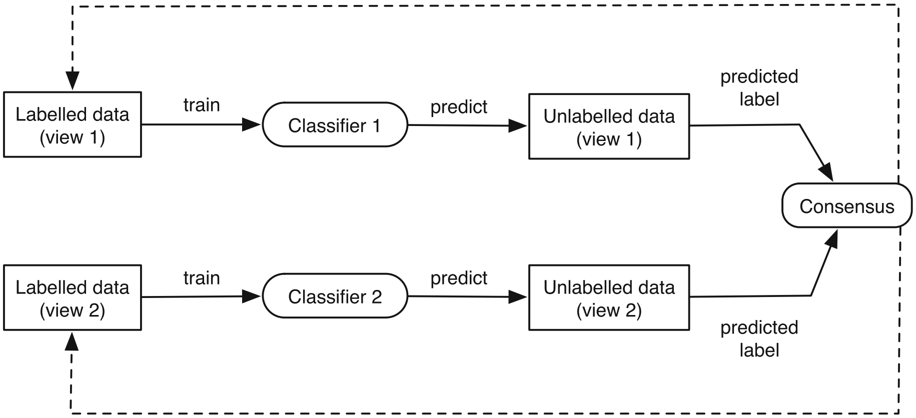


Fig. 3. Classic co-training algorithm.

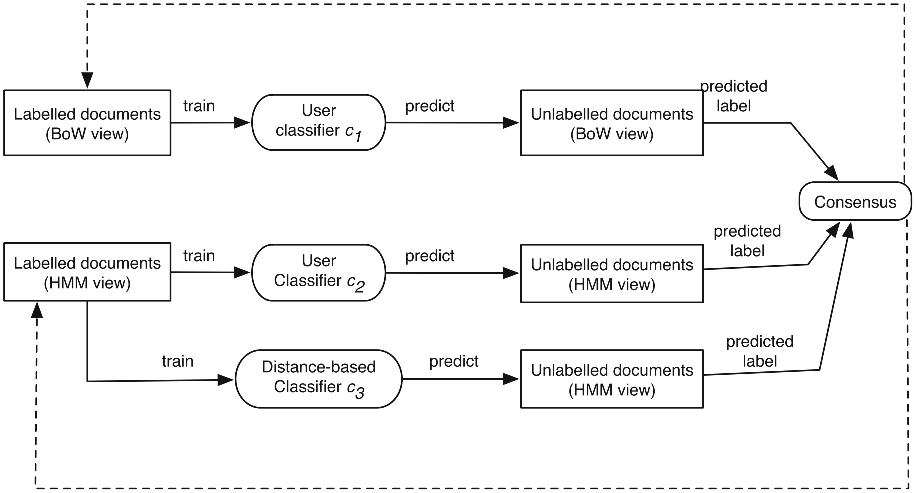

Labelled documents (BoW view)
train
User classifier c_1
predict
Unlabelled documents (BoW view)
predicted label
Consensus
Labelled documents (HMM view)
train
User Classifier c_2
predict
Unlabelled documents (HMM view)
predicted label
train
Distance-based Classifier c_3
predict
Unlabelled documents (HMM view)
predicted label

Fig. 4. Proposed co-training algorithm.

4 Experiments

The goal of the experiments is to test whether the proposed co-training algorithm improves the performance in two document corpora, Reuters and 20 Newsgroup. The collections are described below, and are used in a classification process with an SVM classifier. In order to compare the effectiveness of the proposed algorithm, the two corpora are also classified using the default single-view approach.

The first test collection is the Reuters-21578 document corpus. The document set used for this work contains the documents from the 10 top-sized categories as used in [8], ending up with a total of 8055 documents.

```
**Data**: BoW dataset, classifier model, distance-based classifier, threshold,
      number of iterations
**Result**: Trained classifier
$l_1 \leftarrow$ Labelled portion of dataset with a BoW view;
$l_2 \leftarrow$ Generated labelled portion of dataset with an HMM view;
$u_1 \leftarrow$ Unlabelled portion of dataset with a BoW view;
$u_2 \leftarrow$ Generated unlabelled portion of dataset with an HMM view;
$c_1 \leftarrow$ Classifier model given by the user to train with the BoW view;
$c_2 \leftarrow$ Copy of classifier model given by the user to train with the HMM view;
$c_3 \leftarrow$ Distance-based classifier to improve the consensus;
$t \leftarrow$ Threshold for the distance-based classifier;
$k \leftarrow$ Number of iterations for the co-training algorithm;
$s \leftarrow$ Number of unlabelled documents;
for $i \leftarrow 0$ to $k$ do
    train $c_1$ with $l_1$;
    train $c_2$ with $l_2$;
    train $c_3$ with $l_2$;
    select $s/k$ documents for the iteration;
    $u'_1 \leftarrow$ selected documents from $u_1$;
    $u'_2 \leftarrow$ selected documents from $u_2$;
    for each selected document $d$ do
        $pl_1 \leftarrow$ predict label for $d$ with $c_1$;
        $pl_2 \leftarrow$ predict label for $d$ with $c_2$;
        $pl_3 \leftarrow$ predict label for $d$ with $c_3$;
        $cl \leftarrow$ confidence level for the $pl_3$ prediction with $c_3$;
        if $(pl_1 = pl_2 = pl_3)$ and $(cl >= t)$ then
            $l_1 \leftarrow l_1 \cup d$ from $u'_1$ labelled with $pl_1$;
            $l_2 \leftarrow l_2 \cup d$ from $u'_2$ labelled with $pl_2$;
        end
    end
end
```

Algorithm 1. Co-training algorithm with HMM view.

The second test collection is the 20 Newsgroups dataset. This is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups. It was originally collected by Ken Lang [9] and has become a popular dataset for experiments in text applications of machine learning techniques.

Figure 5 shows the complete experiment process for the evaluation of the proposed co-training algorithm.

Initially, the document corpora need to be pre-processed. Following the BoW approach, we format every document into a vector of feature words in which elements describe the word occurrence frequencies. All the different words that appear in the training corpus are candidates for feature words. In order to reduce the initial feature size, standard text pre-processing techniques are used. A predefined list of stopwords (common English words) is removed from the text, and a stemmer based on the Lovins stemmer [10] is applied. Finally, words occurring in fewer than ten documents of the entire training corpus are also removed.

Once the initial feature set is determined, a dataset matrix is created where rows correspond to documents and columns to feature words. The value of an element in a matrix is determined by the number of occurrences of that feature word (column) in the document (row). This value is adjusted using the TF-IDF statistic in order to measure the word relevance. The application of TF-IDF decreases the weight of terms that occur very frequently in the collection, and increases the weight of terms that occur rarely [11].

Once the pre-processing phase is finished, the corpus is randomly divided into two splits. One of them is considered the labelled set of documents and the other one the unlabelled set of documents after removing their labels. The labelled split is further divided into the Train and Test splits. The Test split contains documents that are reserved to evaluate the performance of the algorithm (see Fig. 5 (a)). Every split (Train, Test, Unlabelled) has one third of the original corpus. In the Reuters dataset, with a total of 8055 documents, each split has 2685 documents. In the 20 Newsgroups dataset, with a total of 18560 documents, each split has 6187 documents.

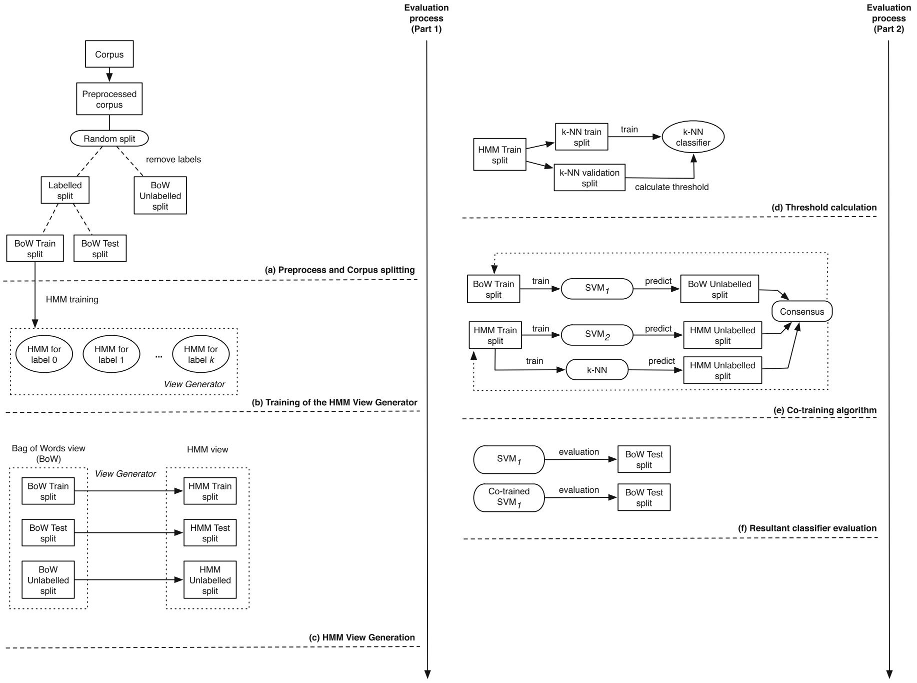


Fig. 5. Experiment workflow for the evaluation of the proposed co-training algorithms.

Using the Train split represented in the bag-of-words approach as input, the View Generator is built training one HMM per label in the corpus (Fig. 5 (b)). The parameterization of the HMM is set with a general approach. The number of states of each HMM is equal to the average number of words in the documents, and the f-factor described in [6] is set to 0.5. Once the View generator is created, it is used to generate the HMM view of each split that is originally represented with the bag-of-words view (see Fig. 5 (c)).

Before the co-training algorithm starts, the threshold for the distance-based classifier must be fixed (Fig. 5 (d)). In this experiment, the selected distance-based classifier is a k-NN with 10 neighbours. When a document is classified with this kind of classifier, the output is not only the predicted label. The classifier also outputs a vector with k elements, where k is the number of possible labels and each element describes the probability of the document having that label. With this information, an average level of confidence can be calculated using the probabilities of correctly assigned labels. In order to do so, the Train split with the HMM view is divided to train a k-NN classifier and calculate the average confidence threshold to label a new document with a validation document set.

Once the threshold is calculated, the co-training algorithm can be executed to improve the training process of a classifier given by the user. In our tests, SVM is selected as the classifier to improve (Fig. 5 (e)). The implementation of the SVM used in this case is LIBSVM [12] and the parameters are those utilized by default in the WEKA environment [13], applying a RBF kernel.

Finally, after applying the co-training algorithm, the resultant co-trained SVM is compared with a trained SVM with a single-view approach using the Test split. This entire process is repeated 50 times for each corpus in order to be able to perform a statistic test on the results.

5 Results

To evaluate the effectiveness of the model, F-measure and Kappa Statistic, evaluation measures commonly utilized in text classification and information retrieval, are used. F-measure is the weighted harmonic mean of recall and precision, and Kappa is a single metric that takes the output confusion matrix of an evaluation and reduces it into one value [14]. Kappa Statistic is interesting because it compares the accuracy of the system to the accuracy of a random system. Possible values of kappa range between -1 (random classification) and 1 (perfect classification).

Table 1 shows the results achieved. The values correspond to the average value achieved in that measure for the total of 50 executions with each method. In addition, the F-measure value corresponds to the weighted average F-measure among all the classes in the corpus. The table compares the results achieved by training the SVM with a single-view BoW approach (called Baseline) as opposed to the results achieved by training the SVM with the proposed co-training algorithm.

Furthermore, in order to demonstrate that the observed results are not just a chance effect in the estimation process, we use a statistical test that gives confidence bounds to predict the true performance from a given test set. A Student's t-test is performed on Kappa and F-measure values achieved in method and corpus.

One test is performed for each pair of collection results achieved in each method. In this case, the baseline scenario (single-view method) is compared in each corpus with the results obtained when applying the co-training algorithm. The difference in a given confidence level is checked to determine if it exceeds the confidence limit stated by a Student's distribution. If so, the *null-hypothesis* (the difference is due to chance) is rejected, proving that the application of the co-training algorithm makes a real difference in the output of the classifier. The values in brackets correspond to the t-value calculated in the comparison of the value collections. t-value gives a measure of how large the difference is between applying and not applying the co-training algorithm: higher absolute values indicate more difference.

According to the results, there is a statistical difference between the baseline and co-training methods, since the calculated t-values outrange the confidence limits. This means that the increase in the average Kappa and F-measure values using the proposed co-training algorithm is significant. The SVM classifier gets an improvement using a multi-view approach, specially in the 20 Newsgroups corpus, because the training pool has increased labelling examples from the unlabelled data, something that is not possible in a single-view approach without overfitting the model. It is important to note that the improvement achieved by using the co-training algorithm laregly depends on the corpus. The quality (in terms of information) of the unlabelled documents influences the whole process. This can be seen in the difference between the results achieved with Reuters and 20 NewsGroups.

Table 1. Results

Corpus	Kappa	F-measure
Reuters		
Baseline algorithm	0,809	0,866
Co-train algorithm	0,811 (−17,59)	0,869 (−19,13)
20 newsgroups		
Baseline algorithm	0,621	0,681
Co-train algorithm	0,667 (−42,27)	0,710 (−31,78)
p-value: 0.05		
t-value (two-tailed, 50 freedom degrees): +/−2.009		

6 Conclusions

In this study, we propose a co-training scheme where an HMM view is used. The proposed view generation algorithm allows any dataset and text classifier given by the user to be applied in a co-training framework and take advantage of a multi-view learning algorithm.

The experimental results show that the application of multi-view techniques improve the accuracy of the text classifiers, especially when the number of documents previously classified (used for training) is far lower than the total number of texts for classification. This is the case when accessing big text corpora as public databases (PubMed, Internet) using semi-automatic searches. As a future work, a bigger study on how different classifiers are affected by the co-training system can demonstrate its usefulness in many scenarios. This implies both the main classifier and the distance-based classifier that is used to control the confidence level of the new view.

Acknowledgements. This work has been funded from the European Union Seventh Framework Programme [FP7/REGPOT-2012-2013.1] under grant agreement n 316265, BIOCAPS, the "Platform of integration of intelligent techniques for analysis of biomedical information" project (TIN2013-47153-C3-3-R) from Spanish Ministry of Economy and Competitiveness and the [14VI05] Contract-Programme from the University of Vigo.

References

1. Blum, A., Mitchell, T.: Combining labeled and unlabeled data withco-training. In: Proceedings of the Eleventh Annual Conference on Computational Learning Theory, COLT 1998, pp. 92–100. ACM, New York (1998)
2. Matsubara, E.T., Monard, M.C., Batista, G.: Multi-view semi-supervised learning: an approach to obtain different views from text datasets. In: Proceedings of the Conference on Advances in Logic Based Intelligent Systems: Selected Papers of LAPTEC 2005, pp. 97–104. IOS Press, Amsterdam (2005)
3. Bickel, S., Scheffer, T.: Multi-view clustering. In: Proceedings of the Fourth IEEE International Conference on Data Mining, ICDM 2004, pp. 19–26. IEEE Computer Society, Washington (2004)
4. Xu, C., Taom, D., Xu, C.: A survey on multi-view learning. CoRR, abs/1304.5634 (2013)
5. Nikolaos, T., George, T.: Document classification system based on HMM word map.In: Proceedings of the 5th International Conference on Soft Computing as Transdisciplinary Science and Technology, CSTST 2008, pp. 7–12, New York, NY, USA, ACM (2008)
6. Vieira, A.S., Iglesias, E.L., Borrajo, L.: T-HMM: a novel biomedical text classifier based on hidden markov models. In: Saez-Rodriguez, J., Rocha, M.P., Fdez-Riverola, F., De Paz Santana, J.F. (eds.) PACBB 2014. Advances in Intelligent Systems and Computing, vol. 294, pp. 225–234. Springer International Publishing, Heidelberg (2014)
7. Rabiner, L.R.: A tutorial on hidden markov models and selected applications in speech recognition. Proc. IEEE **77**(2), 257–286 (1989)
8. Sebastiani, F.: Machine learning in automated text categorization. ACM Comput. Surv. **34**, 1–47 (2002)
9. Lang, K.: Newsweeder: learning to filter netnews. In: Proceedings of the Twelfth International Conference on Machine Learning, pp. 331–339 (1995)

10. Lovins, J.B.: Development of a stemming algorithm. Mech. Transl. Comput. Linguist. **11**, 22–31 (1968)
11. Baeza-Yates, R.A., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley Longman, Boston (1999)
12. Chang, C., Lin, C.: LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. **2**(3), 27: 1–27: 27 (2011)
13. Sierra Araujo, B.: Aprendizaje automático: conceptos básicos y avanzados: aspectos prácticos utilizando el software Weka. Pearson Prentice Hall, Madrid (2006)
14. Viera, A.J., Garrett, J.M.: Understanding interobserver agreement: the kappa statistic. Fam. Med. **37**(5), 360–363 (2005)