Accepted Manuscript

An HMM-based synthetic view generator to improve the efficiency of ensemble systems

L. BORRAJO*, *Department of Computer Science, Higher Technical School of Computer Engineering, University of Vigo, 32004 Ourense, Spain.*

A. SEARA VIEIRA**, *Department of Computer Science, Higher Technical School of Computer Engineering, University of Vigo, 32004 Ourense, Spain.*

E. L. IGLESIAS†, *Department of Computer Science, Higher Technical School of Computer Engineering, University of Vigo, 32004 Ourense, Spain.*

Abstract

One of the most active areas of research in semi-supervised learning has been to study methods for constructing good ensembles of classifiers. Ensemble systems are techniques that create multiple models and then combine them to produce improved results. These systems usually produce more accurate solutions than a single model would. Specially, multi-view ensemble systems 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 algorithm to generate a synthetic view from a standard text dataset. The model generates a new view from the standard bag-of-words approach using an algorithm based on hidden Markov models (HMMs). To show the effectiveness of the proposed HMM-based synthetic view generation method, it has been integrated in a co-training ensemble system and tested with four text corpora: Reuters, 20 Newsgroup, TREC Genomics and OHSUMED. The results obtained are promising, showing a significant increase in the efficiency of the ensemble system compared to a single-view approach.

Keywords: Hidden Markov model, text classification, ensemble systems, multi-view learning.

1 Introduction

Machine learning approaches are widely used for text classification because they are easily trainable and adaptable to different domains and languages. During the past decades two machine learning paradigms, *semi-supervised learning* and *ensemble learning*, have achieved great success.

Semi-supervised techniques, unlike supervised techniques where a large number of labelled examples are required, learn a concept definition by combining a small set of labelled examples and a large set of unlabelled ones [1].

Ensemble systems, also called multiple learning systems, are a popular way of machine learning based on the construction of a set of classifiers whose individual decisions are combined in some way (typically by weighted or unweighted voting) to classify new examples. It has been shown to perform better than the best individual classifier that make them up [2–6]. This is because the ensemble has

*E-mail: lborrajo@uvigo.es
**E-mail: adrseara@uvigo.es
†E-mail: eva@uvigo.es

Vol. 00, No. 0,
doi:10.1093/jigpal/jzz067

greater generalization accuracy that depends on the diversity of each individual classifier as well as on their individual performance, in other words, help minimize incorrect answers generated from individual classifiers in the ensemble by combining the results using different techniques [7].

Multiple learning systems have been applied in many different fields, including text processing. For example, Smith *et al.* [8] combined the results of 19 systems for gene mention recognition in the BioCreative II corpus. They found that the combined system outperformed the best individual system by 3.5 percentage point in terms of F-measure. Baumgartner *et al.* [9] combined three systems for gene name recognition with the same dataset and showed that the performance of the combination increased the F-measure in 3.4 percentage point, the best single system. In the study of Kim *et al.* [10] eight systems for event extraction were combined. The combined system increased by 4 percentage point improvement over the best individual system. Finally, Kang *et al.* [11] combined six publicly available text chunkers using a simple voting approach. As compared to the best single chunker, the F-measure of the combined system improved by 3.1 percentage point for noun phrases recognition and by 0.6 percentage point for verb phrases recognition.

One of the most active areas of research in semi-supervised learning has been to study methods for constructing good ensembles of classifiers [7]. In recent years, a great many ensemble learning systems from multi-view data by considering the diversity of different views have been proposed [12].

In contrast to single-view learning, multi-view learning systems 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.

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 [13–15]. However, despite being more promising than single-view approaches, usually document datasets have no natural multiple views available, so that only one view may be provided to represent the data.

In this work, we propose a hidden Markov model (HMM)-based algorithm to automatically generate a synthetical view from a standard document dataset. Given ba text dataset and a multiple learning semi-supervised system as input, the goal of our algorithm is to improve the system performance by increasing the labelled document pool used to train the classifiers.

The remainder of the manuscript proceeds as follows. The view generation process is described in Section 2, and the method to apply it in an ensemble learning system is presented in Section 3. In Sections 4 and 5 we show the experiments and the results obtained for four different text corpora. Finally, the most relevant conclusions are collected at Section 6.

2 Synthetic 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 end, 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 [16]. 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 Figure 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 classifier, two representations or views of the documents are needed. In this work, a novel synthetic view generator is presented. The model generates a new view from the standard BoW approach using an algorithm based on HMMs.

In a previous study, the authors developed an HMM-based document classifier called T-HMM [17]. 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 [25]. 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 [17, 18]. 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 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.

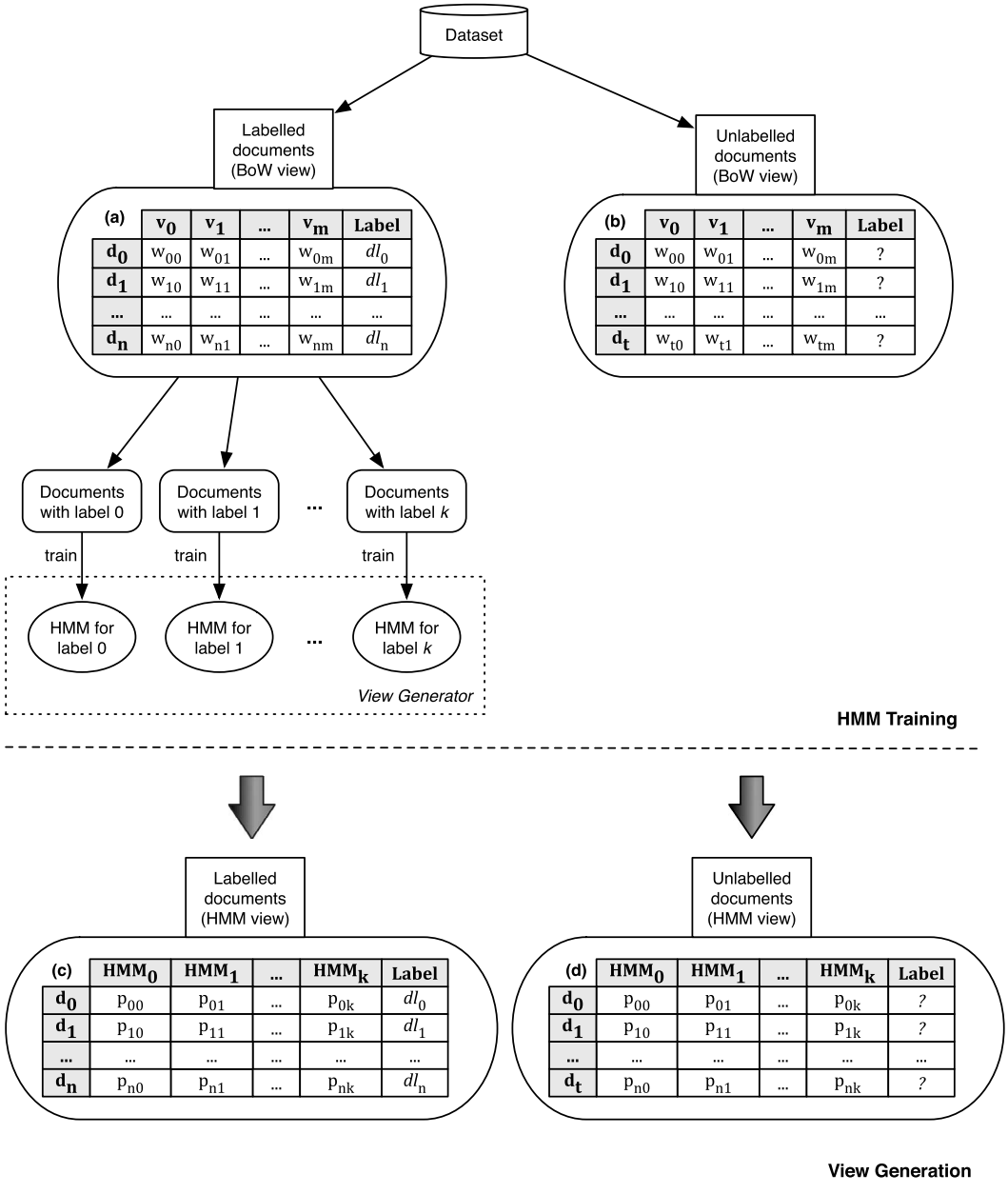
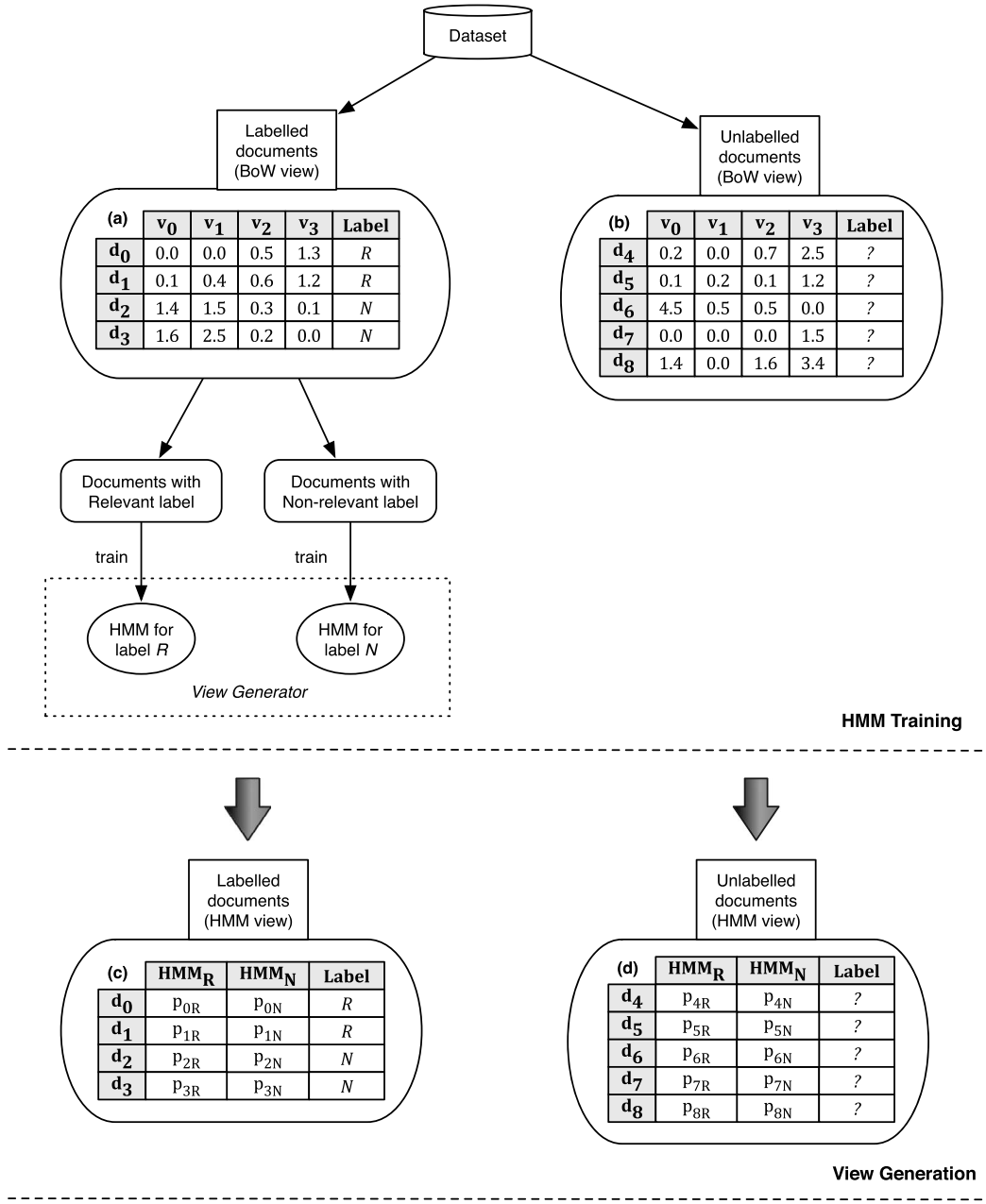


FIGURE 1. Creation of the view generator model and generation of the HMM-view. p_{ij} stands for the probability of the document i being generated by the HMM representing the label j.



(a)	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*

(b)	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	*?*

(c)	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*

(d)	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}	*?*

FIGURE 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.

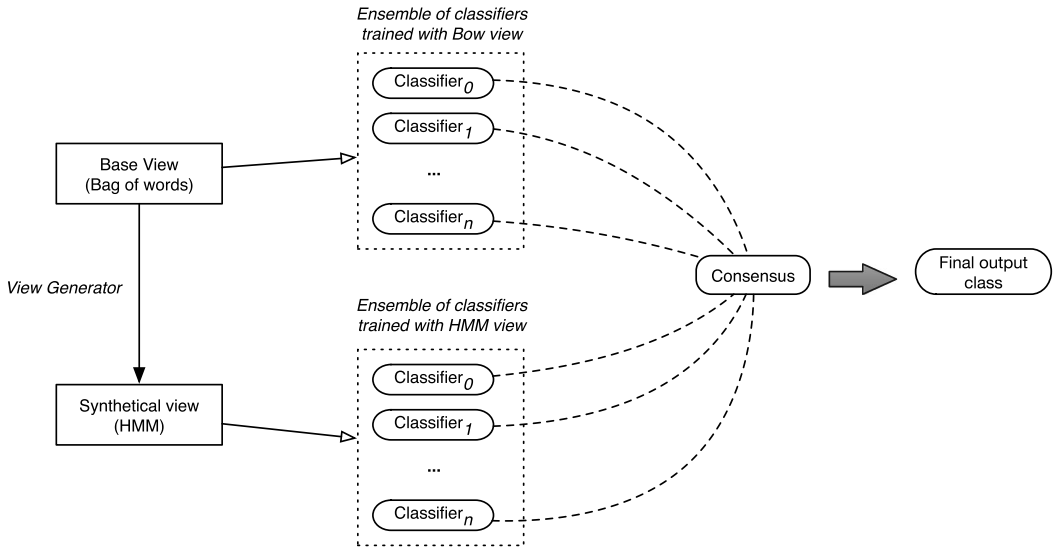


FIGURE 3. Ensemble of classifiers using the synthetical HMM view.

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 transformation process. Labels are only used in the training phase of the HMMs to create the document groups that share the same label.

3 Application of the view generator in an ensemble system

To evaluate the effectiveness of the proposed synthetic view generator, we build an ensemble system that, given a single-view dataset and a set of classifiers provided by the user, increases the performance of the ensemble system by taking advantage of the multi-view classifying process.

Specifically, the proposed framework integrates the classifiers in a co-training-based ensemble system using both a BoW view and an HMM view of the dataset.

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 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 [12].

The classic co-training scheme can be seen in Figure 4. 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.

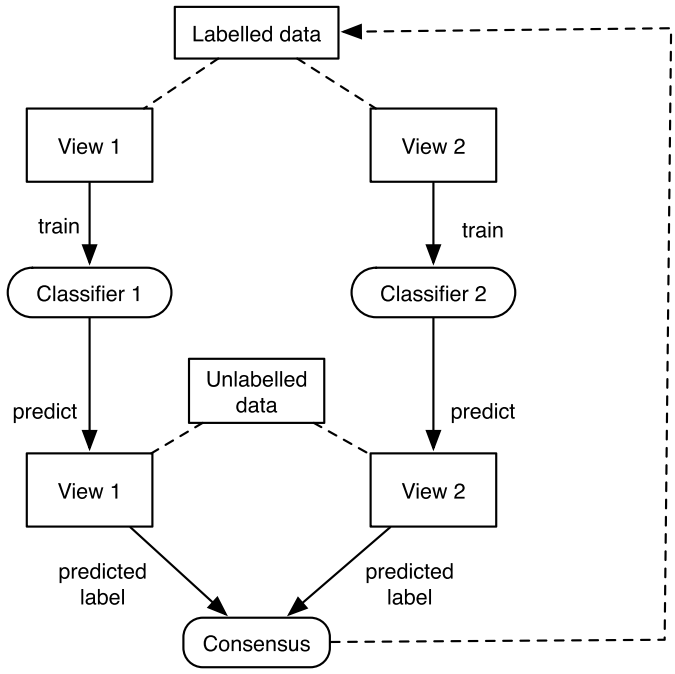


FIGURE 4. Classic co-training algorithm.

Figure 5 shows the proposed co-training scheme. The HMM synthetic 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.

In this example, 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.

4 Experiments

The goal of the experiments is to test the performance of the proposed HMM-based synthetic view generation method using a co-training algorithm as an example of an ensemble of classifiers. The tests are executed with four different document corpora: Reuters, 20 Newsgroup, TREC Genomics, and OHSUMED.

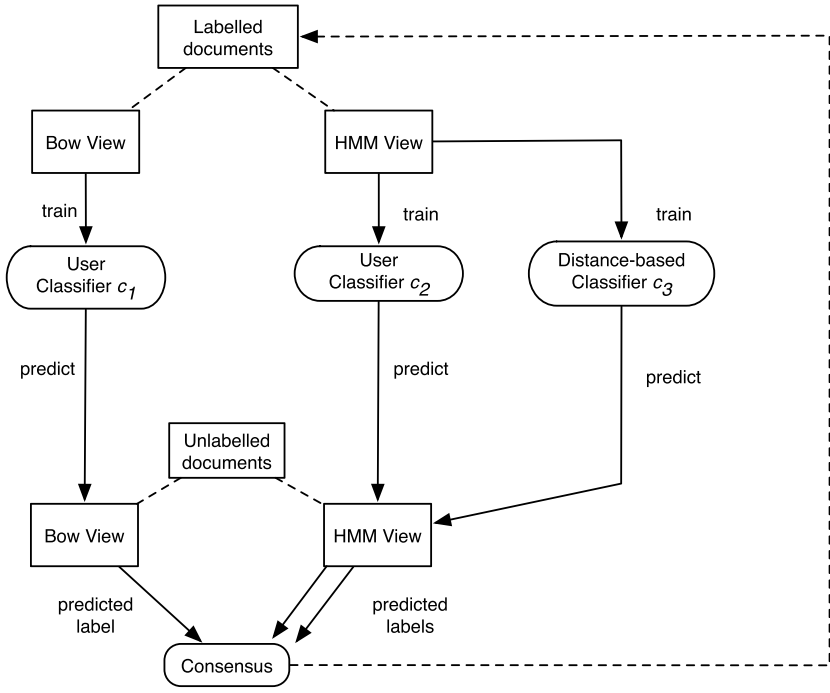


FIGURE 5. Proposed co-training algorithm.

4.1 Datasets

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 [19], ending up with a total of 8055 documents.

The second test collection is the 20 Newsgroups dataset. This is a collection of approximately 20000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups. It was originally collected by Lang [20] and has become a popular dataset for experiments in text applications of machine learning techniques.

The third test dataset is the TREC Genomics dataset. One of the tasks in TREC Genomics 2005 Track [21] was to automatically classify a full-text document collection with the train and test sets, each consisting of about 6000 biomedical journal articles. Systems were required to classify full-text documents from a two-year span (2002–2003) of three journals, with the documents from 2002 comprising the train data, and the documents from 2003 making up the test data. The categorization task assessed how well systems can categorize documents in four separate categories: A (Alelle), E (Expression), G (GO annotation) and T (Tumor). A different corpus is created for each category, where documents can be classified as relevant or non-relevant. In this paper, the collection of abstracts of the Allele corpus is used to test the performance of the proposed system.

The fourth and final test dataset is the TREC Genomics dataset. The OHSUMED test collection, initially compiled by Hersh *et al.* [22], is a subset of the MEDLINE database, which is a bibliographic database of important medical literature maintained by the National Library of Medicine. OHSUMED contains 348566 references consisting of fields such as titles, abstracts and

TABLE 1. Description of the datasets after the preprocessing phase. For each corpus, the table shows the number of documents, number of features, number of classes and the number of documents that belongs to the most common and least common classes.

				Class balance	
Corpus	Documents	Features	Classes	Most common class	Least common class
20 newsgroups	18560	11970	20	996	600
Reuters	8055	3211	8	3916	113
TREC Allele	10795	13578	2	10204	591
TREC GO	10795	13493	2	9933	862
Ohsumed C04	10385	10671	2	7755	2630
Ohsumed C06	9650	10413	2	8430	1220
Ohsumed C14	10580	10661	2	8030	2550
Ohsumed C20	9459	10348	2	8239	1220
Ohsumed C23	10730	10328	2	6778	3952

MeSH descriptors from 279 medical journals published between 1987 and 1991. The collection includes 50216 medical abstracts with an average of 150 words from the year 1991, which were selected as the initial document set. Each document in the set has one or more associated categories (from the 23 disease categories). In order to adapt them to a scheme similar to the TREC corpus, which consists of distinguishing relevant documents from non-relevant ones, we select one of these categories as relevant and consider the others as non-relevant. If a document has been assigned two or more categories and one of them is considered relevant, then the document itself will also be considered relevant and will be excluded from the set of non-relevant documents. Five categories are chosen as relevant: Neoplasms (C04), Digestive (C06), Cardio (C14), Immunology (C20) and Pathology (C23), since they are by far the most frequent categories of the OHSUMED corpus. The other 18 categories are considered as the common bag of non-relevant documents. For each one of the five relevant categories, a different corpus is created in the way mentioned above, ending up with five distinct datasets.

4.2 Evaluation

Initially, each document corpora need to be preprocessed. 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 preprocessing techniques are used. A predefined list of stop words (common English words) is removed from the text, and a stemmer based on the Lovins stemmer [23] is applied. Finally, words occurring in fewer than 10 documents of the entire training corpus are also removed.

When 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 statistc (term frequency–inverse document frequency) 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 [24]. Table 1 shows the characteristics of each corpus after this step.

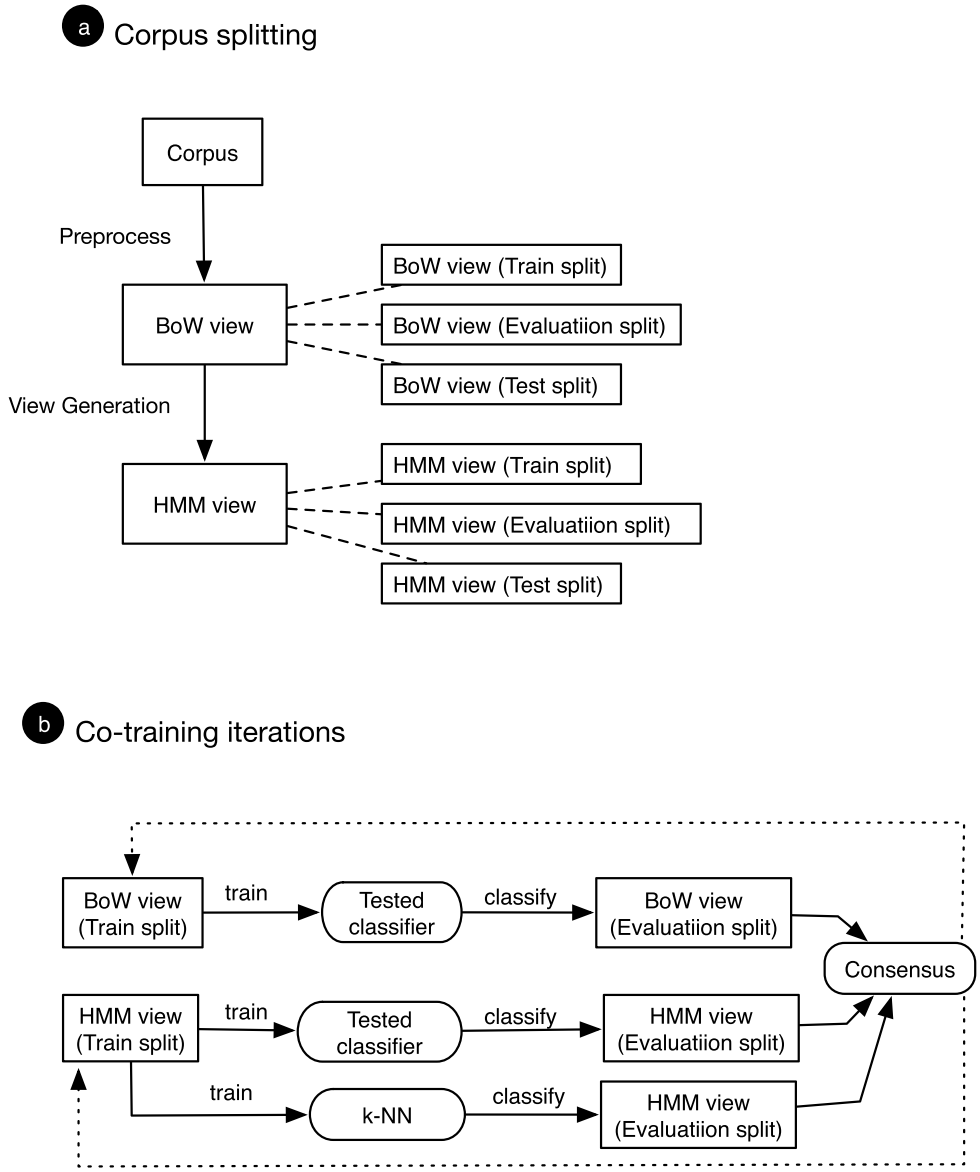


FIGURE 6. (a) Corpus preprocessing and splitting. (b) Workflow for each iteration in the proposed co-training algorithm.

Once the preprocessing phase is finished, the corpus (which is represented in a BoW approach) is randomly divided into three splits: train split, evaluation split and test split (see Figure 6(a)). Using the train split represented in the BoW approach as input, the View Generator is built training one HMM per label in the corpus. 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 [25] 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 BoW view.

Finally, each of the following different experimental setups are executed 10 times in order to test their performance. For each setup, the Support Vector Machine (SVM), Bayes and k-NN classifiers are evaluated with each corpus:

- Evaluate a classifier using the BoW-view train split to train the model and test it with the BoW-view test split (baseline approach).
- Evaluate a classifier using the HMM-view train split to train the model and test it with the HMM-view test split (HMM approach).
- Evaluate a classifier using the proposed co-training algorithm as an example of a ensemble of classifiers. This approach is detailed below.

The proposed co-training algorithm uses the train split of each view as the labelled set of documents, while the evaluation split is considered the unlabelled set of documents after removing their labels. The test split contains documents that are reserved to evaluate the performance of the algorithm (see Figure 6(b)).

The complete specification of the proposed co-training algorithm is detailed in Algorithm 1. The iteration process is depicted in Figure 6(b). In the experiments, the selected distance-based classifier is a *k*-NN with 10 neighbours. When a document is classified with this kind of classifier, the output

Algorithm 1. Proposed Co-training algorithm with HMM view.

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

TABLE 2. Results achieved in every corpus by each classifier/method combination.

Corpus/technique	SVM	Bayes	k-NN	Corpus/technique	SVM	Bayes	k-NN
Reuters				**OHSUMED06**			
Baseline	0,866	0,801	0,616	Baseline	0,815	0,861	0,820
HMM view	0,887 •	0,666 ○	0,874 •	HMM view	0,906 •	0,815 ○	0,909 •
Co-train	0,869 •	0,784 ○	0,614 ∼	Co-train	0,815 ∼	0,871 •	0,820 ∼
20 newsgroups				**OHSUMED14**			
Baseline	0,681	0,545	0,430	Baseline	0,778	0,835	0,539
HMM view	0,784 •	0,086 ○	0,756 •	HMM view	0,886 •	0,721 ○	0,886 •
Co-train	0,710 •	0,545 ∼	0,452 •	Co-train	0,793 •	0,810 ○	0,668 •
Allele				**OHSUMED20**			
Baseline	0,919	0,902	0,919	Baseline	0,812	0,844	0,813
HMM view	0,930 •	0,919 •	0,930 •	HMM view	0,885 •	0,808 ○	0,881 •
Co-train	0,919 ∼	0,908 •	0,919 ∼	Co-train	0,811 ○	0,840 ○	0,814 •
GO				**OHSUMED23**			
Baseline	0,882	0,826	0,882	Baseline	0,492	0,648	0,552
HMM view	0,883 •	0,881 •	0,884 •	HMM view	0,707 •	0,489 ○	0,708 •
Co-train	0,882 ∼	0,828 •	0,882 ∼	Co-train	0,490 ○	0,651 •	0,529 ○
OHSUMED04							
Baseline	0,752	0,836	0,720				
HMM view	0,883 •	0,651 ○	0,882 •				
Co-train	0,757 •	0,828 ○	0,720 ∼				

p-value (confidence level): 0.05 t-value confidence limits (two-tailed, 10 degrees of freedom): $+/-$ 2.2282 *null-hypothesis*: There is no difference between methods •: Method is statistically better than the baseline option ○: Method is statistically worse than the baseline option ∼: There is no statistical difference between method and baseline option

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 used to train a k-NN classifier and calculate the threshold value for the experiments.

As stated earlier, the SVM, Bayes and k-NN classifiers are evaluated in each experimental setup. The implementation of the SVM used in this case is LIBSVM [25] with a Radial Basis Function (RBF) kernel. In addition, all the classifiers use the parameters that are defined by default in the WEKA environment [26], setting the k-NN classifier with three neighbours and the SVM classifier with an RBF kernel with C parameter $= 1$.

5 Results

Table 2 shows the results achieved. The values correspond to the average F-measure value achieved for the total of 10 executions with each method and classifier combination. 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 classifiers with a single-view BoW approach (called baseline) as opposed to the results achieved by training the classifier with the synthetical HMM-view and with the proposed co-training algorithm.

In addition, 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 the collection of F-measures achieved by each pair of methods: (baseline, HMM-view) and (baseline, co-training) with naive Bayes, *k*-NN and SVM classifiers in order to prove their differences. The distance for a given confidence level is checked to determine if it exceeds the confidence limit. In that case, the *null-hypothesis* (the difference is due to chance) is rejected, proving that the model with a higher mean value is statistically better than the other one.

According to the results, the usage of the synthetical view (HMM-view) increases the performance of the SVM and *k*-NN classifiers in all the tested corpus. This is specially relevant in the case of the *k*-NN classifier, which reaches a similar level of accuracy to that of SVM classifier with this view. In the case of the Bayes classifier, the values achieved with the new view are lower in the majority of cases. This is due to the nature of the data in the HMM-view. The numeric values of the synthetical attributes correspond to calculated similarities between clusters of documents and they are expected to work better with function-based classifiers like SVM or k-NN. In general, any function-based ensemble classifier is expected to perform better if it uses the proposed view as a support to the consensus process. It is important to note that the improvement achieved by using this view is superior when using class-balanced corpus like 20 newsgroups, OHSUMED14 and OHSUMED23 (as seen in Table 1). On the rest of the corpus, the increase in performance is less notable, which may indicate a correlation with class balance that should be studied in future works.

Regarding the results achieved using the co-training algorithm, the performance of the classifiers depends largely on the tested corpus. In general, the values remain similar to the baseline approach, showing a visible improvement in the Reuters, 20NewsGroup and OHSUMED14 corpora. It is important to note that the co-training algorithm needs to reach a full consensus in order to add a document to the initial training pool. This implies that only a small percentage of the unlabelled split may be included, reducing the final possible improvement. This is why the performance values are expected to be close to the baseline process.

6 Conclusions

In this study, a new synthetic view generation model that allows any document dataset and text classifier given by the user to take advantage of a multi-view learning approach has been introduced. The model generates a new view from the standard BoW approach using an algorithm based on HMMs.

To show the effectiveness of the proposed HMM-based synthetic view generation method, it has been integrated in an ensemble system and tested with some text corpora.

The experimental results show that, in general, the application of the synthetical view improves the accuracy of the text classifiers. This leads to an improvement of any ensemble of classifiers that uses this new view of the data as a support in the final consensus process.

While the synthetic view generation method produces promising results, further investigations are necessary to ascertain its effectiveness. Specifically, the impact of class balance in the performance of the synthetical view should be studied, as well as the possibility of adding a weight for features and documents in the view generation process. In addition, ensemble systems with different classifiers can be explored. For example, another voting approach can also be incorporated in the final ensemble, constituting another direction for further work.

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] K. Audhkhasi, A. Sethy, B. Ramabhadran and S. S. Narayanan. Creating ensemble of diverse maximum entropy models. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4845–4848, 2012.

[2] R. A. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman, 1999.

[3] W. A. Baumgartner, Z. Lu, H. L. Johnson, J. G. Caporaso, J. Paquette, A. Lindemann, E. K. White, O. Medvedeva, K. B. Cohen and L. Hunter. Concept recognition for extracting protein interaction relations from biomedical text. *Genome Biology*, **9**, S9, 2008.

[4] A. Cano. An ensemble approach to multi-view multi-instance learning. *Knowledge-Based Systems*, **136**, 46–57, 2017.

[5] C. Chang and C. Lin. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, **2**, 227:1–27:27, 2011.

[6] T. G. Dietterich. *Ensemble Methods in Machine Learning*, pp. 1–15. Springer, Berlin, Heidelberg, 2000.

[7] A. Ekbal and S. Saha. Stacked ensemble coupled with feature selection for biomedical entity extraction. *Knowledge-Based Systems*, **46**, 22–32, 2013.

[8] W. Hersh, A. Cohen, J. Yang, R. T. Bhupatiraju, P. Roberts and M. Hearst. Trec 2005 genomics track overview. In *TREC 2005 Notebook*, pp. 14–25, 2005.

[9] W. R. Hersh, C. Buckley, T. J. Leone and D. H. Hickam. Ohsumed: an interactive retrieval evaluation and new large test collection for research. In *SIGIR*, pp. 192–201, 1994.

[10] L. Houthuys, R. Langone and J. A. K. Suykens. Multi-view least squares support vector machines classification. *Neurocomputing*, **282**, 78–88, 2018.

[11] N. Kang, E. M. van Mulligen and J. A. Kors. Comparing and combining chunkers of biomedical text. *Journal of Biomedical Informatics*, **44**, 354–360, 2011.

[12] S. Keretna, C. P. Lim, D. Creighton and K. Bashi Shaban. Enhancing medical named entity recognition with an extended segment representation technique. *Computer Methods and Programs in Biomedicine*, **119**, 88–100, 2015.

[13] J. Kim, T. Ohta, S. Pyysalo, Y. Kano and J. Tsujii. Overview of bionlp'09 shared task on event extraction. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task, BioNLP '09*, pp. 1–9. Association for Computational Linguistics, 2009.

[14] K. Lang. Newsweeder: learning to filter netnews. *In Proceedings of the Twelfth International Conference on Machine Learning*, pp. **331–339**, 1995.

[15] J. B. Lovins. Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, **11**, 22–31, 1968.

[16] E. T. Matsubara, M. C. Monard and G. E. A. P. A. Batista. Multi-view semi-supervised learning: an approach to obtain different views from text datasets. In *Proceedings of the 2005 Conference*

on Advances in Logic Based Intelligent Systems: Selected Papers of LAPTEC 2005, pp. 97–104. IOS Press, Amsterdam, The Netherlands, 2005.

[17] T. Nikolaos and T. George. Document classification system based on HMM word map. In *Proceedings of the 5th International Conference on Soft Computing as Transdisciplinary Science and Technology, CSTST '08*, pp. 7–12. ACM, New York, NY, USA, 2008.

[18] R. Polikar. Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, **6**, 21–45, 2006.

[19] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, **77**, 257–286, 1989.

[20] S. Saha and A. Ekbal. Combining multiple classifiers using vote based classifier ensemble technique for named entity recognition. *Data & Knowledge Engineering*, **85**, 15–39, 2013.

[21] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, **34**, 1–47, 2002.

[22] B. Sierr Araujo. *Aprendizaje automático: conceptos básicos y avanzados: aspectos prácticos utilizando el software Weka*. Pearson Prentice Hall, 2006.

[23] H. Dai, F. Liu, Y. Chen, C. Sun, S. Katrenko, P. Adriaans, C. Blaschke, R. T. Perez, M. Neves, P. Nakov, A. Divoli, M. Mana, J. Mata-Vazquez, and W. J. WilburL. Smith, L. K. Tanabe, R. J. And nee, C. Juo, I. Chung, C. Hsu, Y. Lin, R. Klinger, C. M. Friedrich, K. Ganchev, M. Torii, H. Liu, B. Haddow, C. A. Struble, R. J. Povinelli, A. Vlachos, A. William, L. Hunter, B. Carpenter, R. Tsai, H. Dai, F. Liu, Y. Chen, C. Sun, S. Katrenko, P. Adriaans, C. Blaschke, R. T. Perez, M. Neves, P. Nakov, A. Divoli, M. Mana, J. Mata-Vazquez, and W. J. Wilbur. Overview of BioCreative II gene mention recognition. *Genome Biology*, **9**, S2, 2008.

[24] J. Stiborek, T. Pevn and M. Rehk. Multiple instance learning for malware classification. *Expert Systems with Applications*, **93**, 346–357, 2018.

[25] A. Seara Vieira, E. L. Iglesias and L. Borrajo. T-HMM: a novel biomedical text classifier based on hidden Markov models. In *8th International Conference on Practical Applications of Computational Biology and Bioinformatics (PACBB 2014). Vol. 294 of Advances in Intelligent Systems and Computing*, pp. 225–234. Springer International Publishing, 2014.

[26] C. Xu, D. Tao and C. Xu. A survey on multi-view learning. *CoRR*, **abs/1304.5634** arXiv. 2013. http://arxiv.org/abs/1304.5634.

Received 20 October 2017