From Multiple Views to Single View : A Neural Network Approach

Subendhu Rongali [*]
IBM Research India
subendhu.iitm@gmail.com,

Sarath Chandar A P [*]
IBM Research India
apsarathchandar@gmail.com

Balaraman Ravindran
Indian Institute of Technology
Madras
ravi@cse.iitm.ac.in

ABSTRACT

In most general learning problems, data is obtained from multiple sources. Hence, the features can be inherently partitioned into multiple views or feature sets. For example, a media clip can have both audio and video features. If we concatenate these features to form a single view, we essentially lose some statistical properties exhibited by the views. Since conventional Machine Learning algorithms do not deal with multiple views, Multi-View Learning (MVL) approaches like Co-training and Canonical Correlation Analysis were introduced. In this work, we propose an approach to multi-view learning based on a recently proposed autoencoder model called Predictive AutoEncoder (PAE). Standard PAE works with only two views. We propose ways to generalize the PAE to handle more than two views. Experimental results show that the proposed approach performs better than the existing MVL approaches like co-training and Canonical Correlation Analysis.

Keywords

Multiview Learning, Subspace Learning, Auto-encoder, Neural Network

1. INTRODUCTION

In many areas of scientific analysis, we come across situations where data is gathered from multiple sources. A set of indicators from a given source have specific properties that can be exploited while dealing with that source alone. These sets of indicators are called views. For example, a clipping can be represented by both its audio and video. A document can be represented in multiple languages. An instance can thus have multiple views or in other words, its indicators can be inherently grouped together to form multiple views. Each of these views could be sufficient for classification on their own. They could also be weak. In any case, they provide additional information from the context of other views. Hence, when used together in an ideal setting, it is trivial to assume that the learning model would perform better.

Conventional Machine Learning algorithms like *Decision Trees* [10], *Naive Bayes* [7] or *Support Vector Machines* [4] cannot deal with multiple views. One simple approach to use these algorithms with multi-view data is to concatenate all the views together and consider it as a single view problem. However, this causes over-fitting of data when the training instances are not adequate. It is also not a meaningful approach, since we are not exploiting the statistical properties exhibited by each view.

Multiview Learning [3] is a paradigm in machine learning that aims to achieve optimum results when you have multiple views in the data. Here, the approach involves treating each view separately and using the additional information of the data (view relationships) to jointly optimize the functions on each of the views. This results in better performance as the additional information present here plays a major role in tasks like disambiguation and similarity detection. Several algorithms have been developed for multi-view learning with the prime method being *Co-training* [3]. Co-training works on the principle of consensus. It trains to improve the agreement between the multiple views (typically two). This has spurned off a whole class of algorithms. *Co-EM* [9], which essentially works like Co-training but by assigning probabilistic labels is another popular variant. *Co-Regularization* [13] is a generalized regularization algorithm to deal with multiple views. *Multiple Kernel Learning* [2] has also been adapted to suit the multi-view setting. Here, the kernels correspond to different views and their combination helps in the learning process.

There is another class of algorithms which fall under *Subspace Learning.* These algorithms aim to find a latent subspace where all the views of an instance can be projected. We assume that the views are derived from this latent subspace. Once we have this subspace, we simply run the conventional machine learning approaches for the given task. *Canonical Correlation Analysis (CCA)* [6] is a popular approach in subspace learning. *CCA* and *Kernel CCA (KCCA)* [1] try to maximize the correlation of the projections of different views on a set of basis vectors. The latent subspace typically has fewer dimensions than the views. Hence, these approaches also help in countering the curse of dimensionality.

In this paper, we propose one such subspace learning approach based on neural networks. Our aim is to explore the application of auto-encoders in a multi-view setting. There are some recent development in the neural network commu-

[*] The work was done when the authors were at IIT Madras.

CoDS '14 India

nity to learn common representations for multimodal data [11, 14, 8, 15, 12]. In our work, we specifically attempt to generalize the model proposed in [11], to handle data with more than two views. In [11], the authors use an auto-encoder setup to accomplish Natural Language Processing tasks on bilingual data. They build a shared representation for parallel documents in two different languages and use this rich representation for tasks like cross language document classification and cross language sentiment analysis. The auto-encoder model introduced in [11], Predictive Auto-Encoder (PAE), is the model we have replicated to test on traditional multi-view datasets.

The parallel documents are essentially like multiple views of the semantic equivalent of the document. However, these views are highly correlated. The performance of the PAE in a multi-view setup, where there is no assumed correlation between the views has not been explored. Our goal in this work is to explore the performance of PAE on traditional two-view data and conditionally extend it to cover more views. The major contributions of this paper are the approaches we introduce to extend the PAE to handle data sets with more than two views.

2. BACKGROUND

In this section, we will briefly explain the necessary background to understand the Predictive Auto-Encoder (PAE) and the proposed variants to PAE. An auto-encoder is a three layer neural network consists of an encoder followed by a decoder[5]. The encoder is a function f that maps an input $x \in \mathbb{R}^{d_x}$ to hidden representation $h(x) \in \mathbb{R}^{d_h}$. It can be defined as

$$h(x) = f(x) = s_f(Wx + b_h) \tag{1}$$

where s_f is a nonlinear activation function like sigmoid function.

$$sigmoid(z) = \frac{1}{1+e^{-z}} \tag{2}$$

The parameters of the encoder are a d_h X d_x weight matrix W and a bias vector $b_h \in \mathbb{R}^{d_h}$.

The decoder function g maps the hidden representation h back to a reconstruction y such that,

$$y = g(h) = s_g(W'h + b_y) \tag{3}$$

where s_g is the decoder's activation function, typically either the identity or a sigmoid. The decoder's parameters are the matrix W' and a bias vector $b_y \in \mathbb{R}^{d_x}$. In general, $W' = W^T$.

The auto-encoder is trained to find the parameters $\theta = \{W, b_h, b_x\}$ such that the reconstruction error is minimum. If D_n is the set of training examples, then the objective function to be minimized is given by,

$$\mathcal{J}_{AE}(\theta) = \sum_{x \in D_n} L(x, g(f(x)) \tag{4}$$

where L is the reconstruction error. A typically used reconstruction error function is the Squared error function.

3. PREDICTIVE AUTO-ENCODER

In this section, we describe the Predictive Auto-Encoder model introduced in [11], which is the basis for the proposed multi-view learning approach. This is the basic building block for our proposed model for multiple views.

The PAE consists of an encoder, that maps the two input views into the hidden layer and then a decoder, that reconstructs the two input views. A pictorial representation is given in Figure 1.

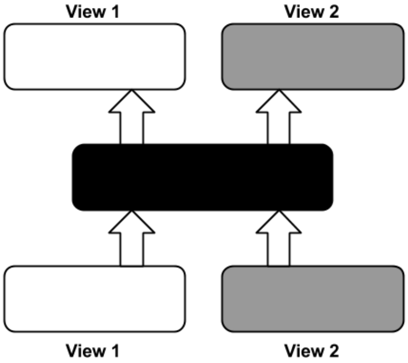

Figure 1: Predictive Auto-encoder for two views

The PAE learns the shared representation of instances in two different views. The procedure consists of two phases. In the first phase, the model takes as input, a parallel list of instances in two different views.

Let $x_i \in R^{V_1}$ be the feature vector of instance i in view V_1 and $y_i \in R^{V_2}$ be the corresponding feature vector in view V_2.

Now, we have a set of parallel instances Z, where

$$Z = \{(x_i, y_i)\}_{i=1}^{n} \tag{5}$$

For a given parallel set of features $z_i = (x_i, y_i)$, we construct two vectors z_i^1 and z_i^2 such that

$$z_i^1 = (x_i \in R^{V_1}, 0) \quad \& \quad z_i^2 = (0, y_i \in R^{V_2}) \tag{6}$$

z_i^1 and z_i^2 are thus representations of x_i and y_i in a $R^{V_1+V_2}$ dimensional space such that the features of the other view are all set to 0. Now, the objective function in the PAE is designed such that we can learn a function $f : R^{V_1+V_2} \rightarrow R^{V_h}$, where R^{V_h} is the hidden layer dimensional space, to make $f(z_i^1)$ and $f(z_i^2)$ highly correlated. Once this is done, to prepare the data for the learning classifier, we simply project the data with two views (x_i, y_i) into the R^{V_h} space and use the resultant vector for the training and testing. We achieve this function f through a PAE in the following way.

Recollecting z_i, z_i^1 and z_i^2 from the beginning, the PAE is now trained to learn the parameters W, b_h and b_y, that minimize the following objective function.

$$\Phi_{PAE}(W, b_h, b_y) = \sum_{i=1}^{n} L(z_i, g(f(z_i^1))) + \sum_{i=1}^{n} L(z_i, g(f(z_i^2))) + \sum_{i=1}^{n} L(z_i, g(f(z_i))) - \alpha \sum_{i=1}^{n} cor(f(z_i^1), f(z_i^2)) \tag{7}$$

Here, L is the reconstruction error and α is the scaling coefficient for the last term.

The conventional auto-encoder objective function consists of only the third term i.e the error in reconstructing features of both views z_i, given features of both views z_i. In this model, the first term represents the reconstruction error while constructing both the views, given just z_i^1 and the second term represents the same value, given just z_i^2. These terms help in improving the knowledge and predictive power of one view about the other view. The final scaled correlation term is introduced to ensure that the hidden representations of both the views are highly correlated. The introduction of this term helps in obtaining a better shared representation.

4. GENERALIZATION OF PREDICTIVE AUTO-ENCODER

The traditional PAE proposed in [11] can be applied for a multi-view problem with two views only. In this section, we propose two different ways to generalize the PAE to handle more than two views. One way is to introduce a variant of the auto-encoder that deals with k views. The other way is to use the described 2-view PAE as a building block and come up with a tree-like framework to handle n views.

4.1 k-PAE Model

In this subsection, we will propose a way to generalize the PAE model to an k-PAE model, where k is the number of views. To generalize PAE, we can concatenate more views to both input and output layer. However, to train such a network, we need to generalize the training objective to handle multiple views.

The reconstruction loss terms in the objective function can be extended in two different ways.

- Loss in reconstructing one view given all other views.
- Loss in reconstructing all the other views given one single view.

This is application dependent and can be selected based on the availability of the views during test time.

While calculating the correlation between the projected representations of the parallel views, PAE uses the Pearson's formula for calculating the covariance. This is defined as the second joint cumulant.

Before describing the way to generalize this cumulant to k views, we will formally define the joint cumulants. The joint cumulant of several random variables $X_1, \dots, X_k$ is defined by a cumulant generating function g and consequently the joint cumulant κ.

$$g(t_1, t_2, \dots, t_k) = logE(e^{\sum_{j=1}^{k} t_j X_j}) \tag{8}$$

$$\kappa(X_1, \dots, X_k) = \sum_{\pi} (|\pi| - 1)!(-1)^{|\pi|-1} \prod_{B \in \pi} E\left(\prod_{i \in B} X_i\right) \tag{9}$$

where π runs through the list of all partitions of $\{1, ..., k\}$, B runs through the list of all blocks of the partition π. The kth cumulant does not directly give us a dimensionless quantity for the k-PAE. But by intuition, we use the normalization factor η, described below. Given $\vartheta(f(z_i^1))$ is the variance of $f(z_i^1)$, for k variables (views), we use the ratio of the kth cumulant to a normalization factor η, where

$$\eta = \sqrt{\vartheta(f(z_i^1)) \cdot \vartheta(f(z_i^2)) \cdots \vartheta(f(z_i^k))} \tag{10}$$

as the equivalent of the correlation coefficient for k variables, which is dimensionless.

In the original two-view PAE, the term $cor(f(z_i^1), f(z_i^2))$ can be defined in terms of joint cumulants as

$$cor(f(z_i^1), f(z_i^2)) = \frac{\kappa(f(z_i^1), f(z_i^2))}{\sqrt{\vartheta(f(z_i^1)) \cdot \vartheta(f(z_i^2))}} \tag{11}$$

Thus, for our k-PAE model, the final term in the optimization function becomes

$$\frac{\kappa(f(z_i^1), f(z_i^2) \cdots f(z_i^k))}{\eta} \tag{12}$$

The objective function of the k-PAE, Φ_{kPAE} thus becomes,

$$\begin{aligned}\Phi_{kPAE}(W, b_h, b_y) = \sum_{i=1}^{n} L(z_i, g(f(z_i^1))) + \cdots \\ + \sum_{i=1}^{n} L(z_i, g(f(z_i^k))) + \sum_{i=1}^{n} L(z_i, g(f(z_i))) \\ -\alpha \sum_{i=1}^{n} \frac{\kappa(f(z_i^1), f(z_i^2) \cdots f(z_i^k))}{\eta}\end{aligned} \tag{13}$$

In this work, we have explored the performance of this generalized model on three views. The equations for this are given below. Figure 2 shows the model.

Given three views,

$$\eta = \sqrt{\vartheta(f(z_i^1)) \cdot \vartheta(f(z_i^2)) \cdot \vartheta(f(z_i^3))} \tag{14}$$

The correlation term in the end comes,

$$\frac{\kappa(f(z_i^1), f(z_i^2), f(z_i^3))}{\sqrt{\vartheta(f(z_i^1)) \cdot \vartheta(f(z_i^2)) \cdot \vartheta(f(z_i^3))}} \tag{15}$$

Hence the optimization function Φ_{3-PAE} for our three view auto-encoder becomes,

$$\begin{aligned}\Phi_{3-PAE}(W, b_h, b_y) = \sum_{i=1}^{n} L(z_i, g(f(z_i^1))) + \sum_{i=1}^{n} L(z_i, g(f(z_i^2))) \\ + \sum_{i=1}^{n} L(z_i, g(f(z_i^3))) + \sum_{i=1}^{n} L(z_i, g(f(z_i))) \\ -\alpha \sum_{i=1}^{n} \frac{\kappa(f(z_i^1), f(z_i^2), f(z_i^3))}{\eta}\end{aligned} \tag{16}$$

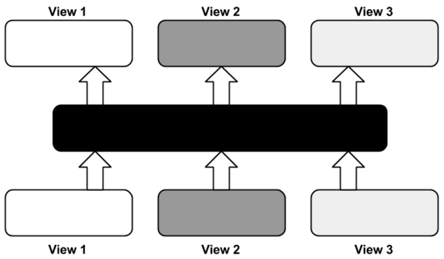

Figure 2: Predictive Auto-encoder for three views

4.2 kc2-PAE Model

In this subsection, we propose a second way to generalize PAE to handle more than two views. In this approach, we use several k-PAEs to handle multiple views. Given k views, we construct $kC2$ 2-PAEs one for each pair of views. Then we learn $kC2$ shared representations one for each pair of views and also train a classifier for each. Then we aggregate the decisions of these $kC2$ classifiers by taking majority voting. This method for 3 views is pictorially depicted in Figure 3.

Even though this model looks like a simple extension of PAE for k-views, in practise, this works extremely well when compared to the $k-$PAE model. We will verify this in the experiments.

Note that this model has advantage in training when compared to the k-PAE model. Unlike k-PAE model, this model does not require a parallel data which has all k views at a time. Each of the 2-PAE model in the setting can be trained separately with different amount of training data and we require not more than a pair of views for each training instance. Even during testing, we can use only a set of classifiers out of the given pool of classifiers based on the availability of views.

5. EXPERIMENTS AND RESULTS

We initially test the performance of the two-view PAE on traditional multiview dataset. This is to support our intuition to use the PAE framework in a non-NLP setting, where it was initially proposed. We then conduct the experiments for the proposed models for three views.

5.1 Datasets Description

5.1.1 WebKB Dataset

This dataset consists of academic web pages collected from computer science department web sites at four universities: Cornell, University of Washington, University of Wisconsin, and University of Texas. These pages can be grouped into six classes: student, staff, faculty, department, course and project. There are two views containing the text on the page and the anchor text of hyperlink respectively.

For our experiment, we have extracted the instances of two classes - course and faculty. The feature set size in each view was the top 10k words in each set. We performed PCA on these views for dimensionality reduction and arrived at 253 features each.

5.1.2 Amazon Multilingual Dataset

This is the dataset used in [11]. The dataset consists of 50k reviews, each in English and French. The ratings were on a scale of 1 to 5 and each of them had 10k documents. For our experiment, we grouped reviews with ratings 1 and 2 as negative sentiment reviews and those with 4 and 5 as positive sentiment reviews. We used all the 50k reviews for training the PAE as that phase is unsupervised. However, for the actual classification, we used only these 40k reviews i.e 20k negative class reviews and 20k positive class reviews. The number of features in each view were 10k.

5.1.3 RCV Multilingual Dataset

The Reuters RCV Multilingual dataset consists of 6 samples of 1200 documents, balanced over 6 labels - E21, CCAT, M11, GCAT, C15 and ECAT. Each sample is made of 5 views. The documents are present in five different languages - English, French, German, Italian and Spanish. The documents were initially in english and they were machine-translated to obtain the remaining four views. The features in each view are 2000 words, selected by the k-medoids algorithm.

For our experiment, we have chosen all the documents of two classes - E21 and CCAT in three languages - English, French and German. We thus have 1200 E21 samples and 1200 CCAT samples.

5.2 Performance on Amazon Multilingual dataset

Amazon Multilingual dataset has been used in [11] to prove the efficiency of PAE over other cross lingual approaches in NLP like translate-and-train and translate-and-test. In this experiment, we wanted to compare the performance of PAE with the standard Multiview Learning approaches like co-training and Canonical Correlation Analysis.

Note that this dataset is not a typical multiview learning dataset since both the views are higly correlated. Also both are strong views i.e each sufficient for good classification results on their own. We learnt 40 dimensional representation for the data when using CCA and PAE. The classifier used in all the experiments was Gaussian Naive bayes. The results for this data set are as given in Table 1.

Table 1: Amazon Review Results - Accuracy

Model	Accuracy
Canonical Correlation Analysis	0.70
Co-training	0.61
2-PAE	**0.72**

From the table, it is evident that PAE performs better than co-training. CCA is close to PAE in performance. But it is important to note that CCA is not scalable for huge data while PAE is clearly scalable due to the usage of stochastic gradient descent with mini-batches for training. We do not need to load the entire data into memory at any point of time.

5.3 Performance on the WebKB dataset

Previous experiment verified the superiority of PAE over standard multi-view learning appraoaches in a dataset with highly correlated views. However, in a traditional multi-view setting, the views will be less correlated and even most of the views will be weak.

In this experiment, we consider one such traditional multi-view dataset - the WebKB dataset. In this experiment also, we learnt 40 features using both CCA and PAE. The classifier used here is Decision Tree Classifier. The results for this data set are as given in Table 2.

Table 2: WebKB results - F Measure

Model	F1-measure
Canonical Correlation Analysis	0.60
Cotraining	0.57
PAE	**0.68**

Even in a dataset where one of the views is weak (hy-

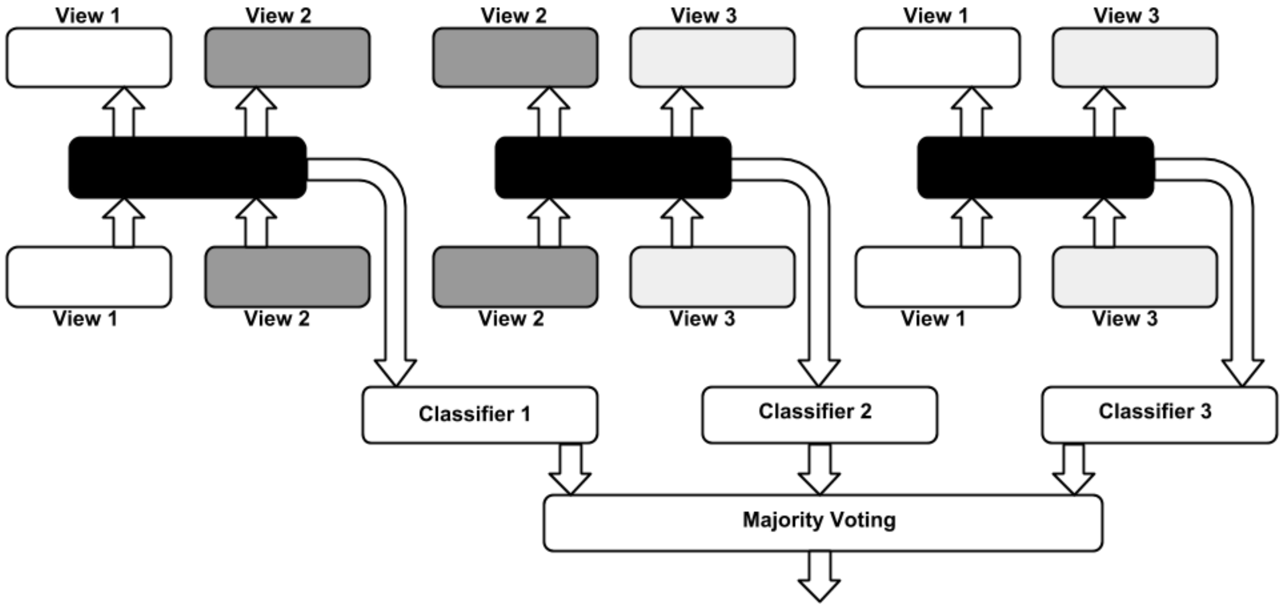

Figure 3: 3C2 Model for three views

perlink view), it is found that PAE performs much better than CCA and co-training. Also, the difference between the performance of CCA and PAE is larger in this data. This illustrates that PAE works well even with weak views.

5.4 Performance on RCV Multilingual Dataset

In the previous two experiments, we demonstrated the performance of PAE on two view datasets. We will now demonstrate the performance of the proposed k-PAE models with RCV Multilingual dataset. This dataset has three views: English, German and French. We use a simple concatenation approach as our first baseline. Here, the features if all the three views are concatenated to form a standard machine learning problem with one view. We use the three-view co-training approach as our second baseline. The third baseline is the majority voting among k models each trained with one of the k views.

5.4.1 3-PAE Model Results

In 3-PAE model there are a lot of ways to get the final features that are used for classification. In the two-view PAE, we projected the composite vector consisting of features of both views into the shared space for the new features. This was just plain intuition that more information would yield better results. In this case we can either project features of one, two or all three views to get the shared representation features for an instance. For this, we make a composite vector of all three views and set the features of the views we want to project to their actual values and those of the views we don't want to project to 0. So, the vector for only English would be $(v_e^1, \cdots, v_e^n, 0, \cdots, 0, 0, \cdots, 0)$ and that of English-German would be $(v_e^1, \cdots, v_e^n, 0, \cdots, 0, v_g^1, \cdots, v_g^n)$. The results for all ways of projecting features of the three languages into the shared space are tabulated. The results for the complete projection of the features of all three languages are tabulated in the end.

It is interesting to see here that the English-German and French-German projections performed better than the complete feature projection. But we can't attribute this to the strength of the German features because we can see that the German-only projection performed poorly. The results are however not completely unexpected. The model seems to be doing a fair job and it would be interesting to test it on more datasets. It is observed that co-training with 3 views performs poorly.

5.4.2 3C2 Model Results

In this experiment, we first tested for individual views. Classifiers were built for only English, only French and only German features. Table 4 shows the results. We then tested for the 3C2 model. The results for the 3C2 model are also tabulated. Rows 4,5 and 6 show the results of the two-view PAE on each pair of languages. The majority voting for the 3C2 model was taken on the results of these three classifiers.

We can observe that the proposed model performs better than the simple concatenation approach and those with individual views. We have chosen these as our baseline strategies since there are no standard algorithms for three views. We cannot apply CCA for 3 views. However, due to the nature of the dataset i.e being made through machine translation rather than manually, there isn't a great deal of extra information between views. This, we believe, has effected the performance of the basic PAE units. Also, the dataset needs to be bigger to enable the PAEs to learn more meaningful shared representations.

6. CONCLUSIONS

In this work, we have explored how the Predictive Autoencoder introduced in [11] performs in a multiview setting. It has performed better than two standard state-of-the-art approaches - Co-training and CCA on the WebKB and the Amazon reviews datasets.

We have also proposed an extension to this two-view PAE model to cover datasets with n views. We tested the two

Table 3: Performance of 3-PAE in RCV Multilingual dataset

Model	Precision	Recall	F1-Score
Projecting only English features	0.75	0.75	0.75
Projecting only French features	0.74	0.74	0.74
Projecting only German features	0.73	0.73	0.73
Projecting English and German features	0.78	0.78	0.77
Projecting English and French features	0.73	0.72	0.73
Projecting French and German features	0.78	0.78	0.78
Projecting features from all languages	0.75	0.75	0.75
Simple Concatenation - Baseline 1	0.77	0.77	0.77
Co-training - Baseline 2	0.59	0.59	0.58
Voting - Baseline 3	**0.81**	**0.81**	**0.81**

Table 4: Performance of 3C2-PAE in RCV Multilingual dataset

Model	Precision	Recall	F1-Score
English View only	0.75	0.75	0.75
French View only	0.77	0.77	0.77
German View only	0.74	0.74	0.73
Eng-Ger Shared Representation	0.75	0.75	0.75
Eng-Fre Shared Representation	0.76	0.76	0.76
Fre-Ger Shared Representation	0.75	0.75	0.75
3C2 Model	**0.82**	**0.82**	**0.82**
Simple Concatenation - Baseline 1	0.77	0.77	0.77
Co-training - Baseline 2	0.59	0.59	0.58
Voting - Baseline 3	0.81	0.81	0.81

proposed approaches on a derived RCV dataset with three views and the 3C2-PAE model performs quite well. The 3-PAE model fared decently, but it still needs to be improved to form a reliable method.

It would also be interesting to test the proposed n-view PAE in the context of Natural Language Processing tasks dealt with in [11], from where the original PAE model was taken. Given sufficient manually processed data in multiple languages, we could build a massive shared representation for all the languages together. This model would then be able to deal with any of the languages over a wide range of applications. The shared representation would be an interesting semantic space for a host of languages.

7. REFERENCES

[1] S. Akaho. A kernel method for canonical correlation analysis. *arXiv preprint cs/0609071*, 2006.

[2] F. R. Bach, G. R. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the smo algorithm. In *Proceedings of ICML*, page 6, 2004.

[3] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM, 1998.

[4] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[5] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[6] H. Hotelling. Canonical correlation analysis (cca). *Journal of Educational Psychology*, 1935.

[7] D. D. Lewis. Naive (bayes) at forty: The independence assumption in information retrieval. In *Machine learning: ECML-98*, pages 4–15. Springer, 1998.

[8] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal deep learning. In *Proceedings of ICML*, pages 689–696, 2011.

[9] K. Nigam and R. Ghani. Analyzing the effectiveness and applicability of co-training. In *Proceedings of the ninth international conference on Information and knowledge management*, pages 86–93. ACM, 2000.

[10] J. R. Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[11] A. P. Sarath Chandar, M. M. Khapra, B. Ravindran, V. Raykar, and A. Saha. Multilingual deep learning. *Deep Learning Workshop NIPS*, 2012.

[12] A. P. Sarath Chandar, S. Lauly, H. Larochelle, M. M. Khapra, B. Ravindran, V. Raykar, and A. Saha. An autoencoder approach to learning bilingual word representations. In *Proceedings of NIPS*, 2014.

[13] V. Sindhwani, P. Niyogi, and M. Belkin. A co-regularization approach to semi-supervised learning with multiple views. In *Proceedings of ICML Workshop on Learning with Multiple Views*, pages 74–79. Citeseer, 2005.

[14] N. Srivastava and R. Salakhutdinov. Multimodal learning with deep boltzmann machines. In *NIPS*, pages 2231–2239, 2012.

[15] Y. Zheng, Y.-J. Zhang, and H. Larochelle. Topic modeling of multimodal data: an autoregressive approach. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE*, 2014.